Tutorial on Autonomous Vehicles Technologies for Perception & Decision-Making

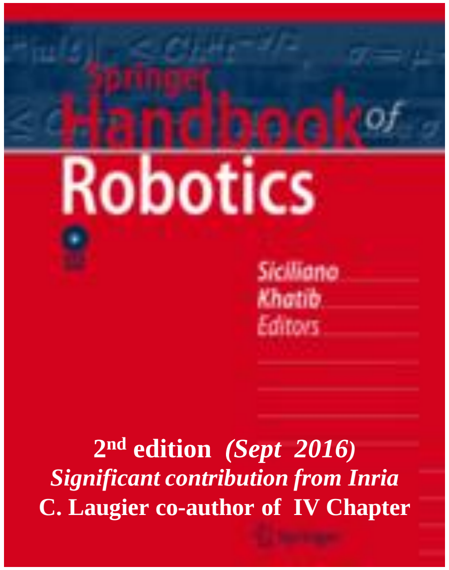
Christian Laugier

Tutorial @ Institut d'Automne en IA 2018 (IA2 2018)
Campus ISAEE-Supaero, Oct 15-19 2018

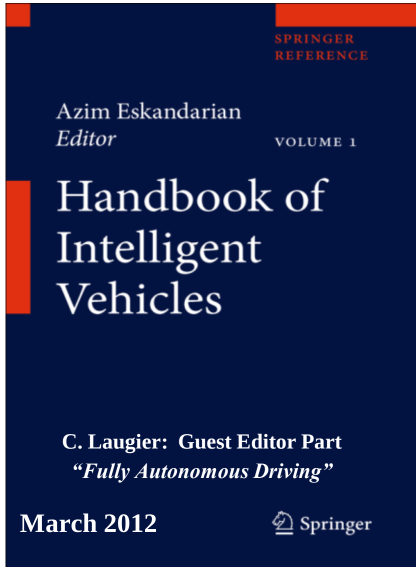
Autonomous Vehicles Technologies for Perception and Decision-Making

Dr HDR Christian LAUGIER
Research Director at Inria & Scientific Advisor for Probayes and for Baidu China
Inria Grenoble Rhône-Alpes, Chroma team
Christian.laugier@inria.fr

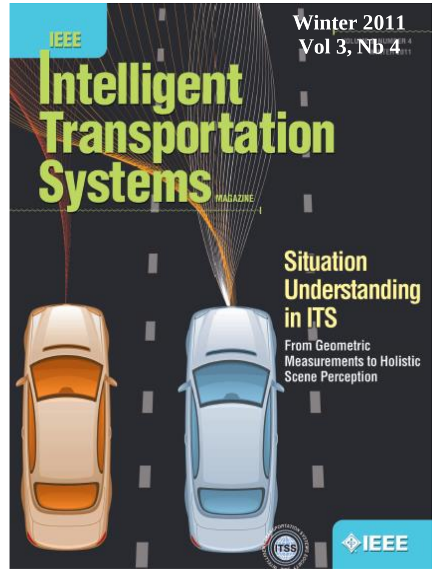
informatics mathematics



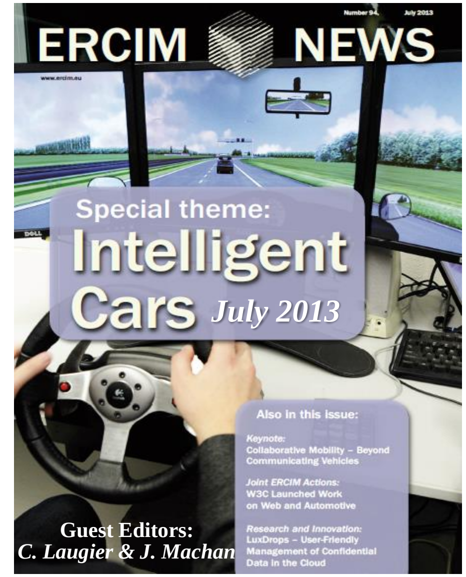
2nd edition *(Sept 2016)*
Significant contribution from Inria
C. Laugier co-author of IV Chapter



C. Laugier: Guest Editor Part
"Fully Autonomous Driving"
March 2012

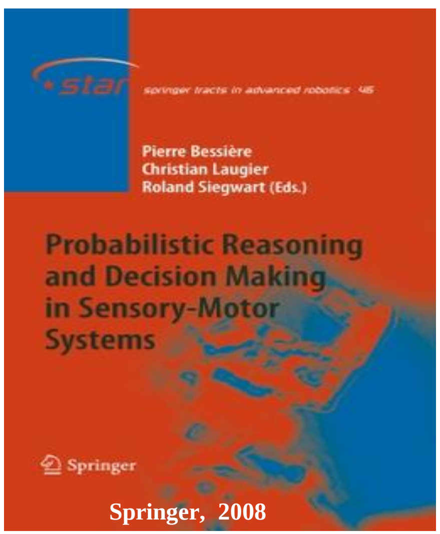


Winter 2011
Vol 3, Nb 4



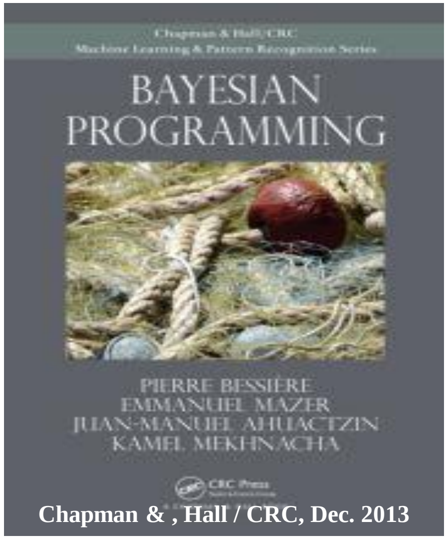
July 2013
Guest Editors:
C. Laugier & J. Machan

Relevant Literature on Robotics & IV & ITS



Springer, 2008

IEEE RAS Technical Committee on "AGV & ITS"
Chairs: Ph. Martinet, C. laugier, C. Stiller
Numerous Workshops & Journal Special issues since 2002
=> Membership open



Chapman & , Hall / CRC, Dec. 2013

Content of the Tutorial

- ❑ **Socio-economic & Technological Context + State of the Art**
- ❑ Decisional & Control Architecture – Outline
- ❑ Bayesian Perception *(key Technology 1)*
- ❑ Embedded Bayesian Perception & Experimental results
- ❑ Bayesian Risk Assessment & Decision-making *(Key Technology 2*

Cars & Human Mobility
A current Psychological & Technological breakthrough

A quick on-going change of the role & concept of **private car** in human society !

Last century *=> Ownership & Feeling of Freedom*
Affective behaviors & Shown Social position
Driving pleasure ... but less and less true !

Next cars generation *=> Focus on* ***Technologies*** *for*
Safety & Comfort & Reduced Pollution
Driving Assistance v/s Autonomous Driving

Context

- *Expected 3 Billions vehicles & 75% population in cities in 2050 => Current model not scalable !*
- *Accidents: ~1.2 Million fatalities/Year in the world => No more accepted !*
- *Driving safety & Nuisance issues (pollution, noise, traffic jam, parking …) are becoming* ***a major issue for Human Society & Governments & Industry***

Cars & Human Mobility
A current Psychological & Technological ... *gh*

A quick on-going change of the role & concept of **private** ...

Last century => *Ownership & Feelin*...
Affective behaviors & Shown ...
Driving pleasure ... *but l*...

...*eration* => *Focus on* ***Technologies*** *for*
...*fety & Comfort & Reduced Pollution*
...*riving Assistance v/s Autonomous Driving*

Towards less ownership & More shared mobility systems
&
Increased Autonomy

- Technolog... change the **mobility habits** of people
  => Sha... *rpooling, more ADAS & Autonomy (e.g. Tesla autopilot)*
  => ... *ber, BlaBlaCar, Robot Taxis (Uber, Nutonomy)*…

...arket for Automotive Industry
...*2012 & Expected* ***$261 billions*** *in 2020* [(f)]

[(f)] Fo... ...obal Market for ADAS Systems by 2020. ABI Research. 2013.

Autonomous Cars – *State of the Art (~ 30 years R&D)*

- **Some early results in Europe** *(80's & 90's)*

1986 VaMors (Dickmann, Munich U)
=> First autonomous vehicle on a road (mainly based on CV)
=> Followed by EU project Prometheus

Pioneer work at INRIA in the 90's
- ✓ *Autonomous parking*
- ✓ *Platooning in cities*
- ✓ *People mover (Cycab)*

Autonomous Cars – *State of the Art (~ 30 years R&D)*

- **Some international important Events & Results** *(first decade 21st century)*

2004& 2006 Darpa Grand Challenges
High speed & Off-road
Significant step towards Motion Autonomy
… But still some uncontrolled behaviors in 2004 !!!

2007 Darpa Urban Challenge
97 km, 50 manned & unmanned vehicles, 35 teams
… Impressive progress towards motion autonomy, but still some collisions (decision-making mainly based on automatom)

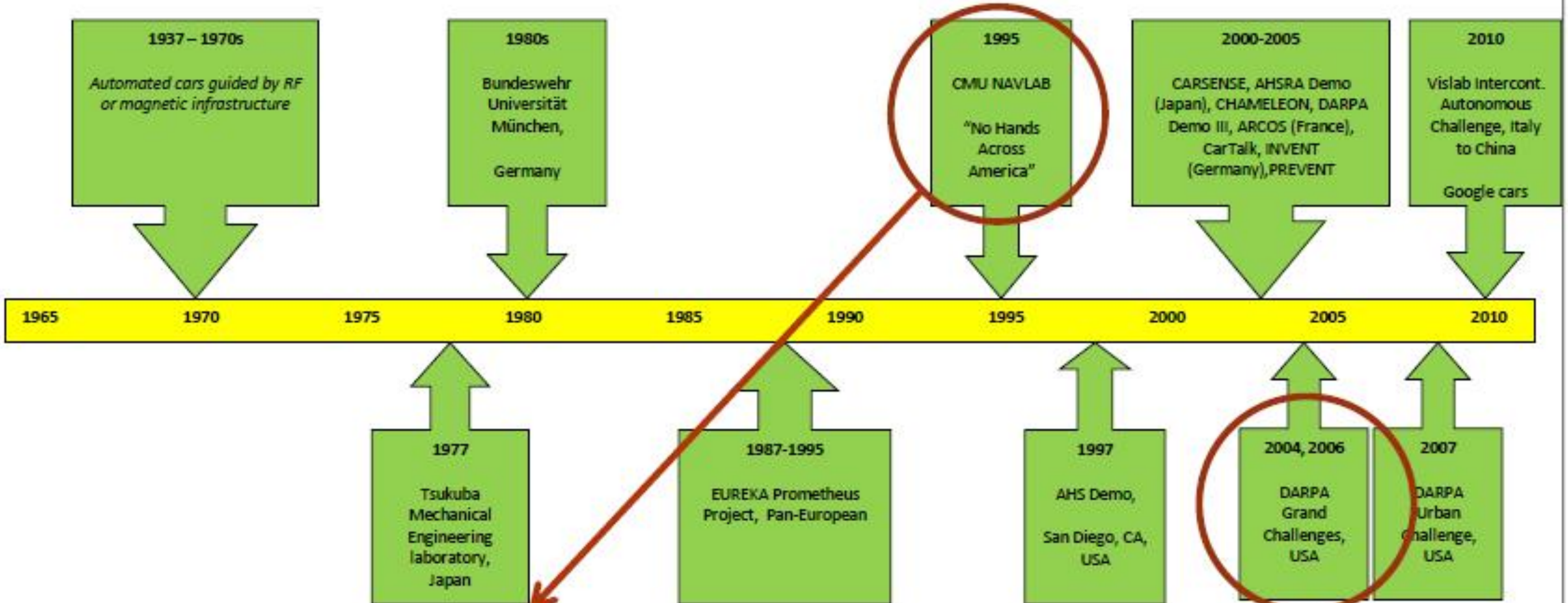
2010 VIAC Intercontinental Autonomous Challenge
13 000 km covered, 3 months race, leader + followers (A. Broggi)
=> See Spring 2011 IEEE RAM issue for more details

2011 Google Car project
Fleet of 6 automated Toyota Prius, equipped with a costly 3D lidar (dense mapping)
140 000 miles covered on California roads with occasional human interventions

Autonomous Cars – *State of the Art (~ 30 years R&D)*

Credit to Paul E. Rybski, CMU (2010)
The path to Commercial Autonomous Cars: The Darpa Urban Challenge and Beyond

Brief Timeline of Autonomous Cars



NAVLAB No Hands Across America (2005) : Highway driving

DARPA Grand Challenge (2004, 2006): High-speed offroad driving

http://www.cs.cmu.edu/~red/Red/redteam.html
http://cs.stanford.edu/group/roadrunner//old/announcements.html

http://www.cs.cmu.edu/afs/cs/project/alv/www/

http://www.youtube.com/watch?v=xkJVV1_4l8E

Autonomous Cars & Driverless Vehicles

- Strong involvement of Car Industry & Large media coverage
- An expected market of 500 B€ in 2035
- Technologies Validation & Certification *=> Numerous recent & on-going real-life experiments + Simulation & Formal methods (e.g. EU Enable-S3 2016-19)*

Tesla Autopilot based on Radar & Mobileye
Commercial ADAS product

3D Lidars & Dense 3D mapping
Numerous vehicles & Millions of miles driven

Mobility Service

Cybus experiment, La Rochelle 2012
=> *CityMobil Project & Inria*

Drive Me trials (Volvo, 2017)
- *100 Test Vehicles in Göteborg, 80 km, 70km/h*
- *No pedestrians & Plenty of separations between lanes*

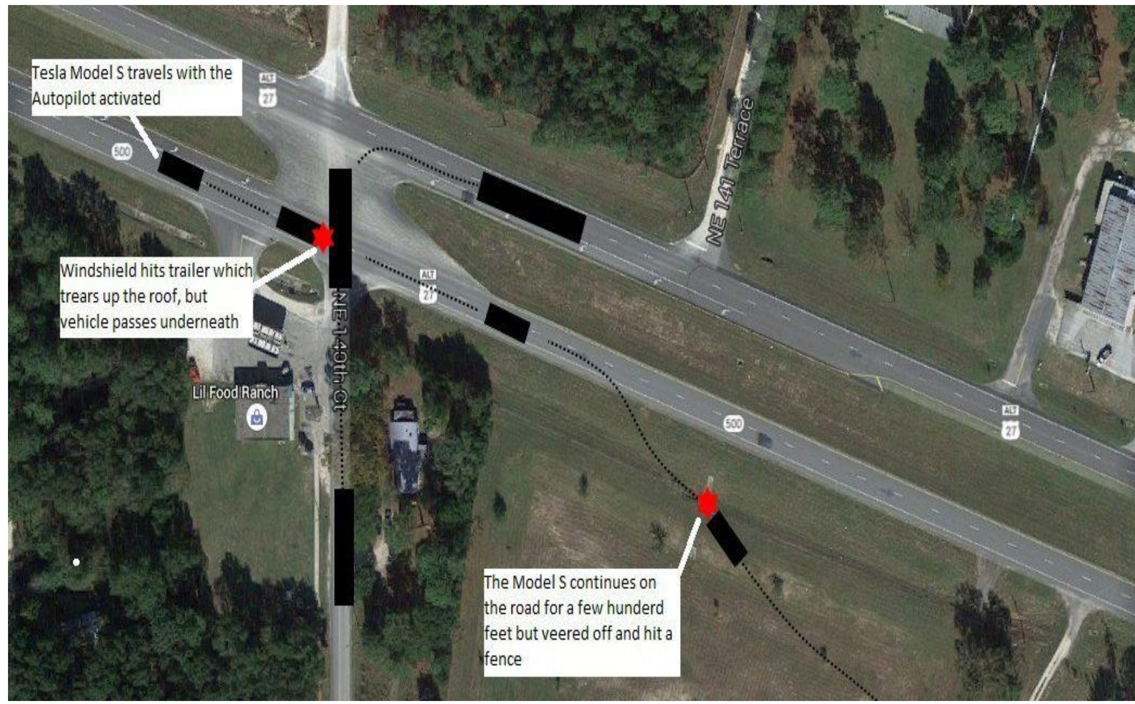
Robot Taxi testing in US *(Uber, Waymo)* & Singapore *(nuTonomy)*
*=> **Mobility Service**, Numerous Sensors … Safety driver in the car during testing*

Millions of miles driven (Tesla, Waymo, Uber…)
Several benign & serious accidents in past few years
Safety is still not guaranteed !

Safety issues: *Example of the Tesla accident (May 2016)*

- **Safety is still insufficient** *(a false sense of Safety for users ?)*
  - *=> Still some Perception & Situation Awareness errors (even with commercial systems)*
  - *=> On May 7th 2016,* ***Tesla driver killed*** *in a crash with Autopilot active (and driver not attentive)*

Tesla Model S – Autopilot
Front perception:
Camera (Mobileye)+ Radar + US sensors



Autopilot didn't detected the trailer as an obstacle *(NHTSA investigation + Tesla conjecture)*

- **Camera** => *White color against a brightly lit sky ?*
- **Radar** => *High ride height of the trailer probably confused the radar into thinking it is an overhead road sign ?*

Safety issues: *Example of the Uber Accident (March 2018)*

- **Self-driving Uber kills a woman in first fatal crash involving pedestrian**
  Temple, Arizona, March 2018

- **The vehicle was moving at 40 mph and didn’t reduced its speed before the crash (collision risk not detected). The Safety Driver didn’t reacted**

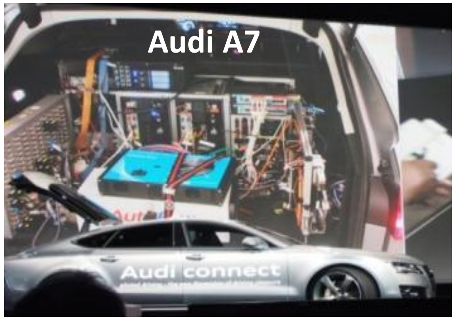
- **In spite of the presence of multiple onboard sensors (several lidars in particular), the perception system didn’t predicted the collision !**

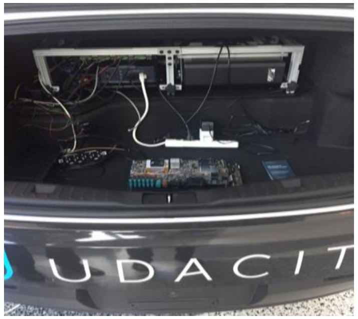
Perception: State of the Art & Today's Limitations

- Despite significant improvements during the last decade of both Sensors & Algorithms, **Embedded Perception** is still one of the major bottleneck for Motion Autonomy

  => Obstacles detection & classification errors, incomplete processing of mobile obstacles, collision risk weakly address, scene understanding partly solved…

- **Lack of Robustness & Efficiency & Embedded integration** is still a significant obstacle to a full deployment of these technologies





Lack of Robustness & Efficiency

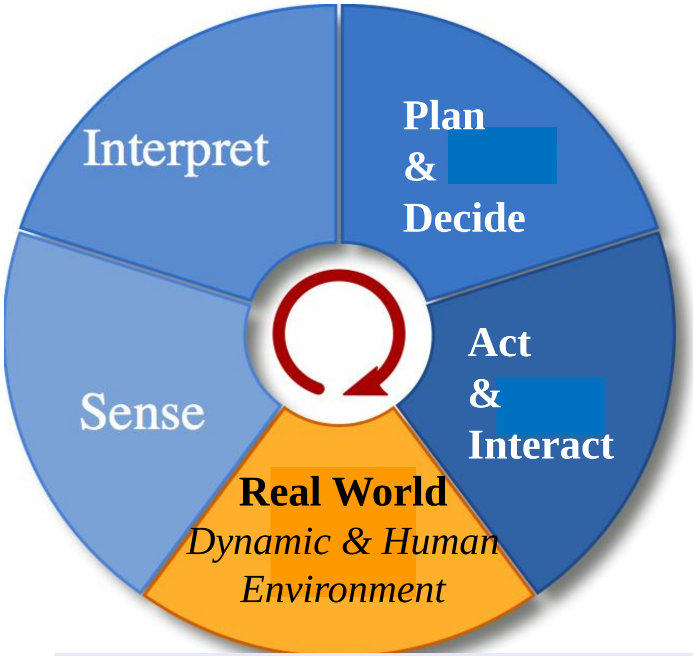
Lack of Integration into Embedded Sw/Hw

- Until recently, car trunks was most of the time full of electronics & computers & processor units
- On-board high computational capabilities & dedicated softwares are still required, even if new products currently appear on the market (e.g. Nvidia Drive-PX, Ambarella embedded vision platform ..)

Content of the Tutorial

- ❑ Socio-economic & Technological Context + State of the Art
- ❑ **Decisional & Control Architecture – Outline**
- ❑ **Bayesian Perception** ***(key Technology 1)***
- ❑ Embedded Bayesian Perception & Experimental results
- ❑ Bayesian Risk Assessment & Decision-making *(Key Technology 2*

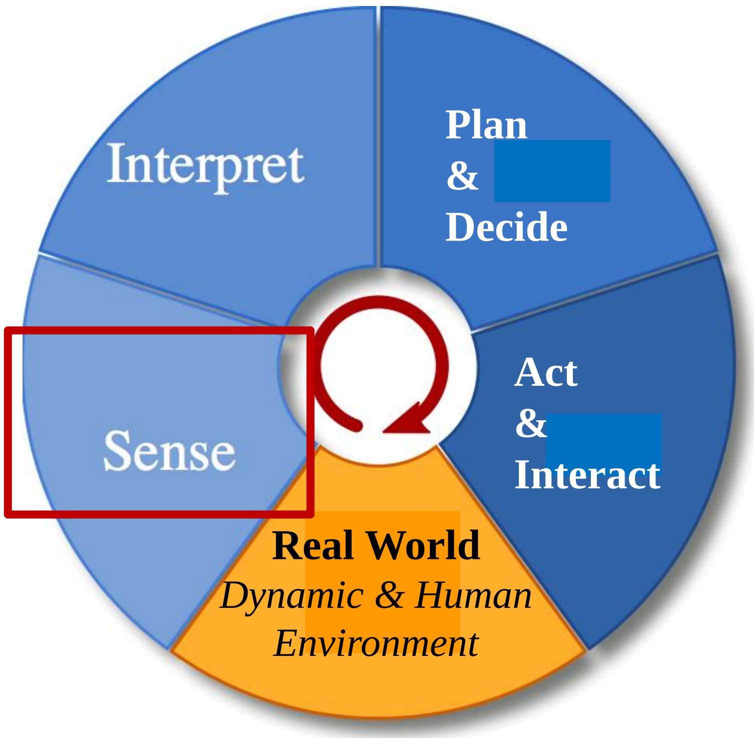
Decisional & Control Architecture – *Outline*



- ❑ **How to control Robot actions in a Dynamic world populated by Human Beings (under strong real-time constraints) ?**

- ❑ **Combining & Adapting interdependent functions for:**
  - ✓ Sensing the environment using various sensors
  - ✓ Interpreting the dynamic scene (using Semantics and Prior Knowledge)
  - ✓ Planning Robot motions & Deciding of the most appropriate action to be executed
  - ✓ Acting & Interacting in the real world (Safety & Acceptability)

Decisional & Control Architecture – *Sensing*



❑ Objective

Perceive what is happening in the Dynamic Scene using various sensors

❑ Main Difficulty

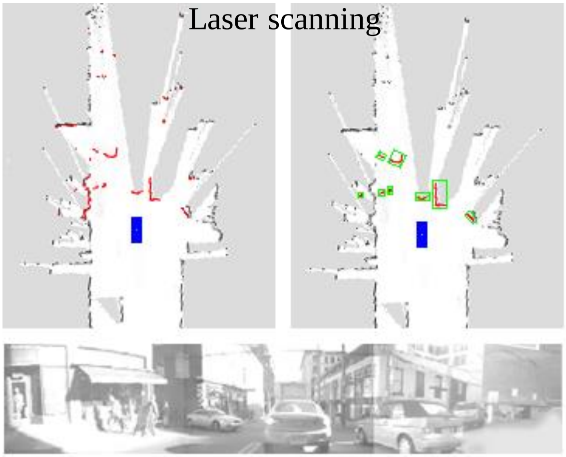
- ✓ Huge heterogeneous sensory data
- ✓ Sensing errors & Uncertainty
- ✓ Real-time processing

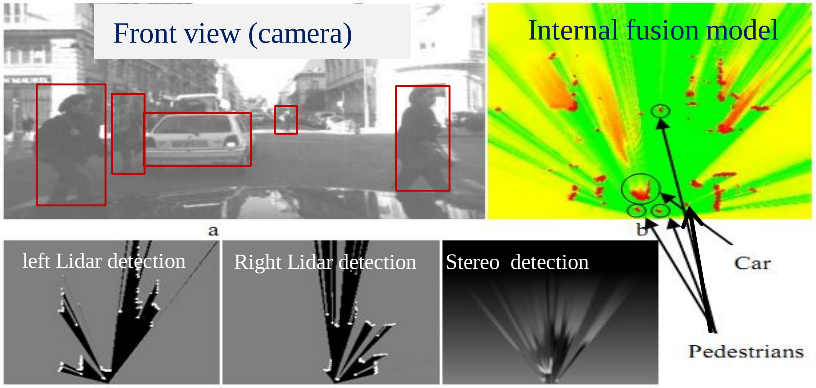
❑ Main Functions

- ✓ Localization & Mapping (SLAM)
- ✓ Static Obstacles + Mobile Objects Detection & Tracking

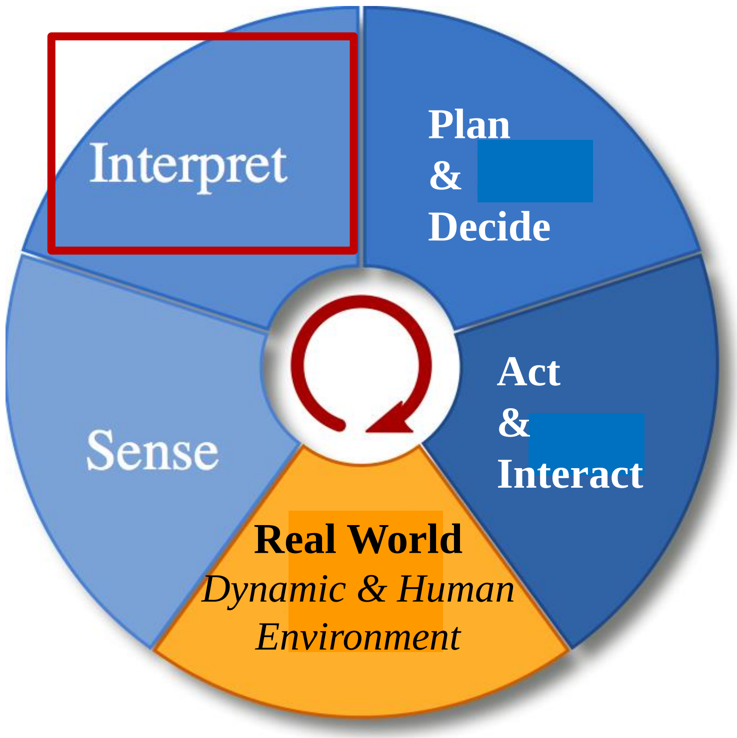
❑ Main Models & Algorithms

- ✓ Bayesian Filtering
- ✓ Feature based & Grid based approaches





Decisional & Control Architecture – *Scene Understanding*

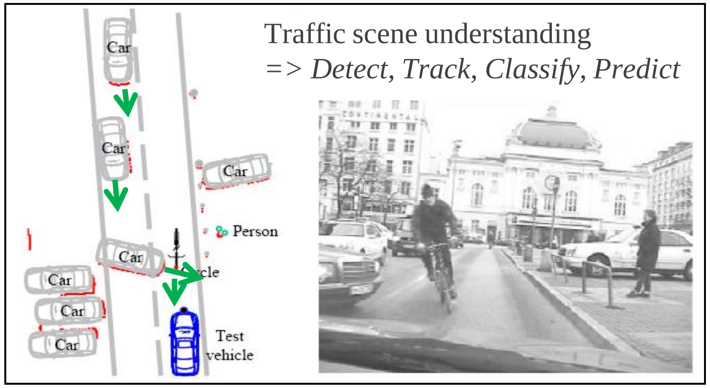


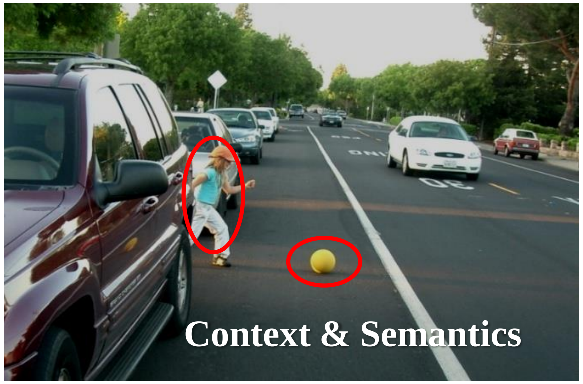
Objective

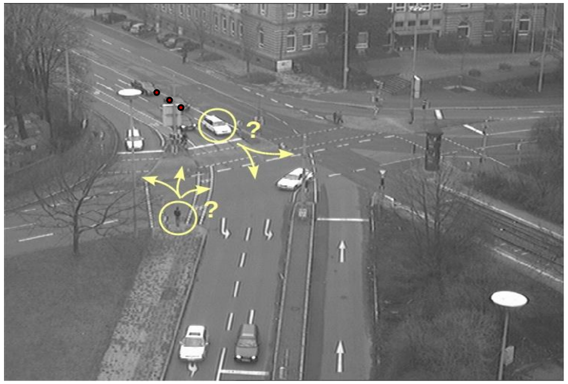
Understand the content of the Dynamic Scene using **Contextual & Semantic knowledge**

Main Difficulty

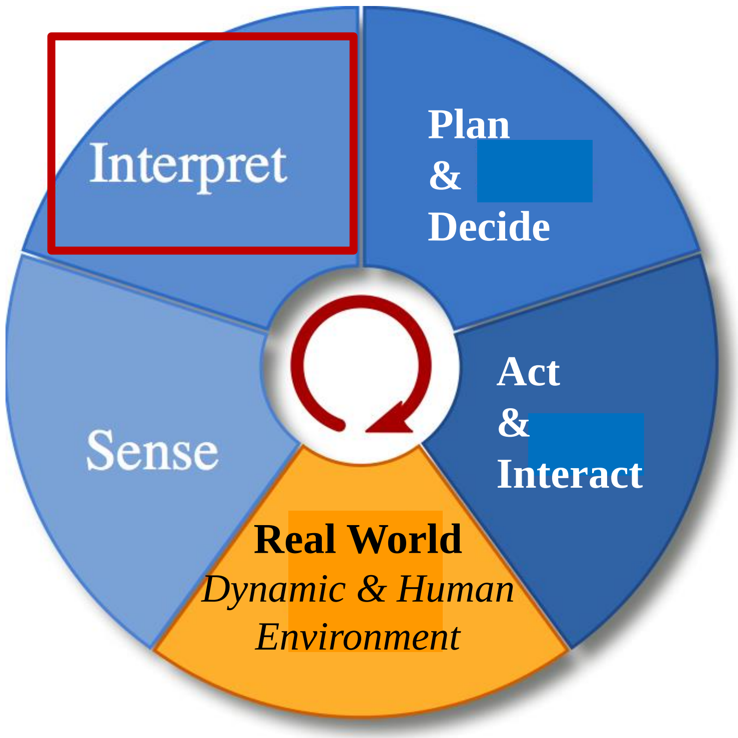
- ✓ Uncertainty
- ✓ Real-time processing
- ✓ Reasoning about various type of knowledge (history, context, semantics, prediction)







Decisional & Control Architecture – *Scene Understanding*

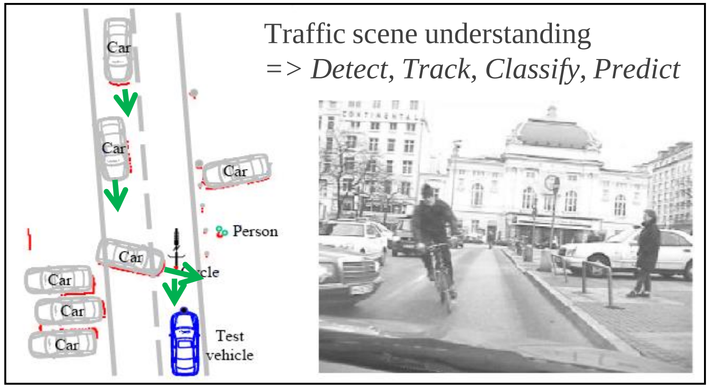


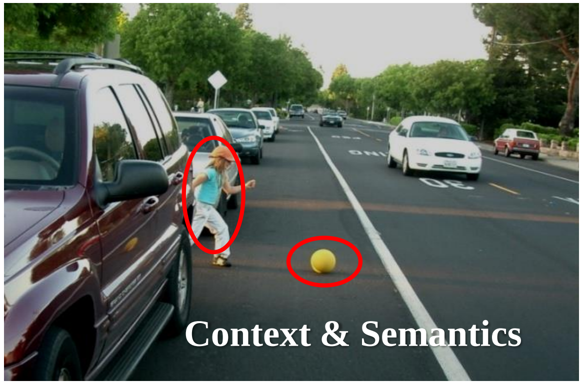
❑ Main Functions

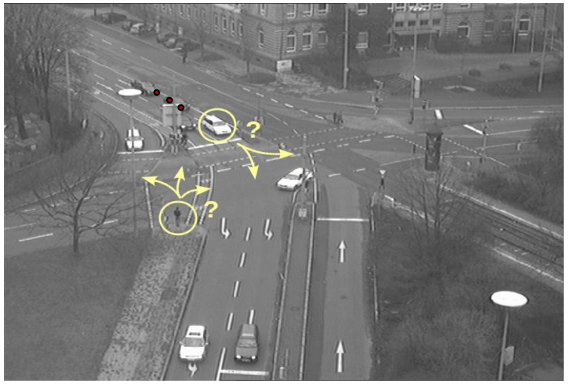
- ✓ Detection & Tracking of Mobile Objects (DATMO)
- ✓ Objects classification (recognition)
- ✓ Prediction & Risk Assessment (avoiding future collisions)

❑ Main Models & Algorithms

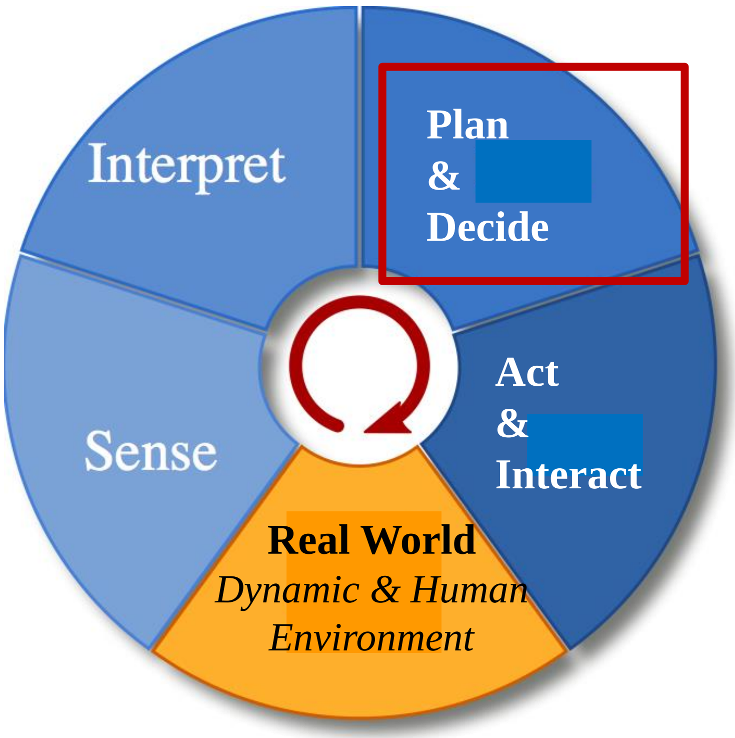
- ✓ Bayesian Perception Paradigm
- ✓ Behaviors modeling & learning
- ✓ Bayesian approaches for Prediction & Risk Assessment







Decisional & Control Architecture – *Decision-making*



❑ Objective

Planning robot motions & Deciding of the most appropriate action to be executed by the robot (Goal & Context & Risk)

❑ Main Difficulty & Functions

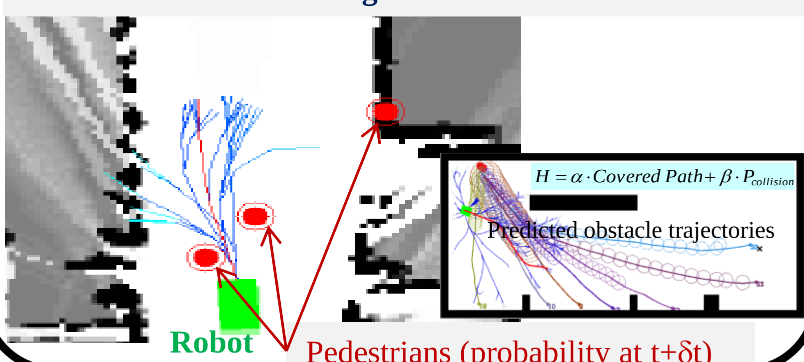
✓ On-line Motion Planning under various constraints (time, kinematic, dynamic, uncertainty, collision risk, social)

✓ Decision making under uncertainty using contextual data (history, semantics, prediction)

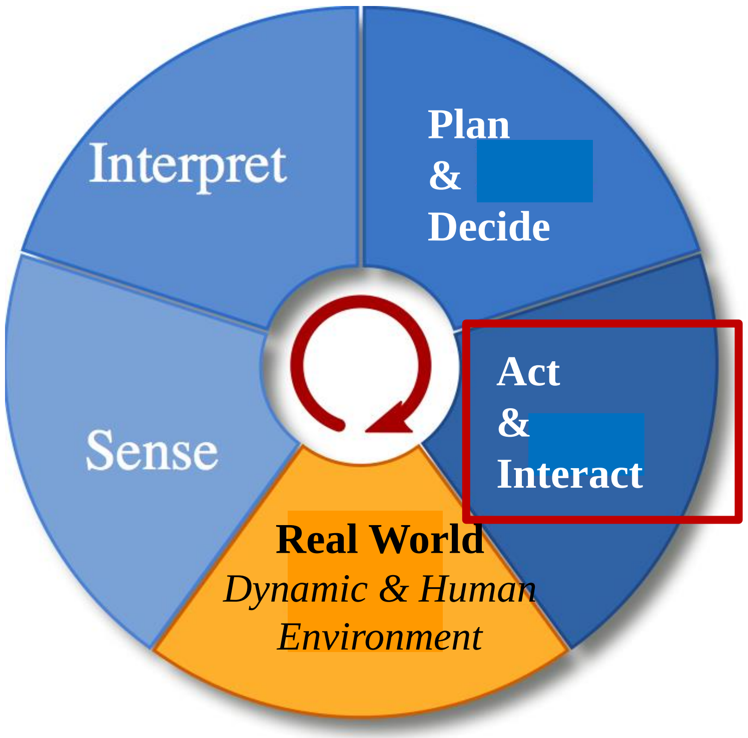
❑ Main Models & Algorithms

✓ Iterative Risk-based Motion Planning (e.g. Risk-RRT)
✓ Decision making using Contextual data & Bayesian networks

Iterative Motion Planning under Time & Risk constraints



Decision-making (avoiding collision)



❑ Objective

Controlling the robot for executing **Safe & Socially Acceptable robot actions**, while taking into account the related **Human – Robot Interactions**

❑ Main Difficulty & Functions

✓ Robot navigation while taking into account both Safety & Social constraints
✓ **Human in the loop**

❑ Main Models & Algorithms

✓ Human-Aware Navigation paradigm (safety & social filters)
✓ Intuitive Human-Robot Interaction

Content of the Tutorial

- ❑ Socio-economic & Technological Context + State of the Art
- ❑ Decisional & Control Architecture – Outline
- ❑ **Bayesian Perception** ***(key Technology 1)***
- ❑ Embedded Bayesian Perception & Experimental results
- ❑ Bayesian Risk Assessment & Decision-making *(Key Technology 2*

Embedded Perception: Main features

Complex Dynamic Scenes understanding

Situation Awareness & Decision-making

Dealing with unexpected events
e.g. Road Safety Campaign, France 2014

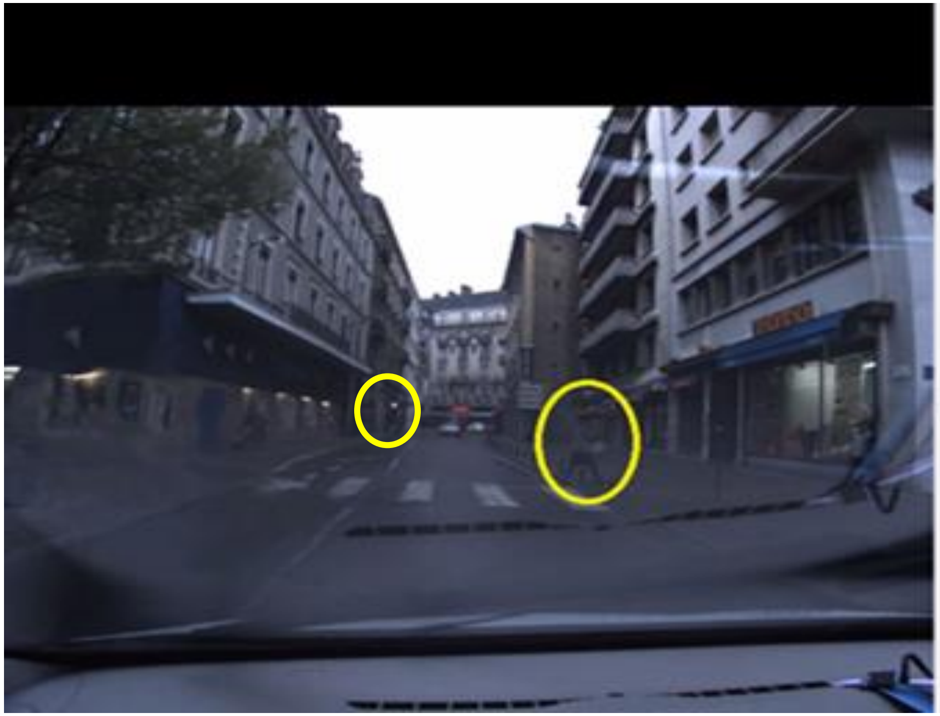
Anticipation & Prediction
for avoiding upcoming accidents

Main features

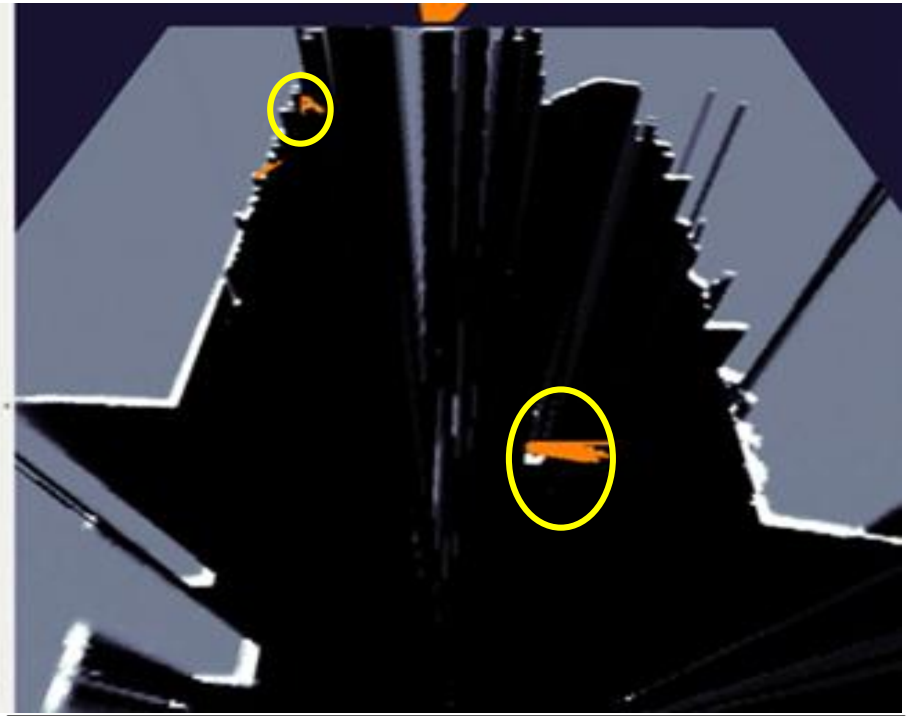
- ✓ Dynamic & Open Environments *=> Real-time processing*
- ✓ Incompleteness & Uncertainty *=> Appropriate Model & Algorithms (probabilistic approaches)*
- ✓ Sensors limitations *=> Multi-Sensors Fusion*
- ✓ Human in the loop *=> Interaction & Behaviors & Social Constraints (including traffic rules)*
- ✓ Hardware / Software integration *=> Satisfying Embedded constraints*

Improving robustness using Multi-Sensors Fusion

Camera Image at Dusk *(Pedestrians not detected)*



Processed Lidar data *(Pedestrians detected)*



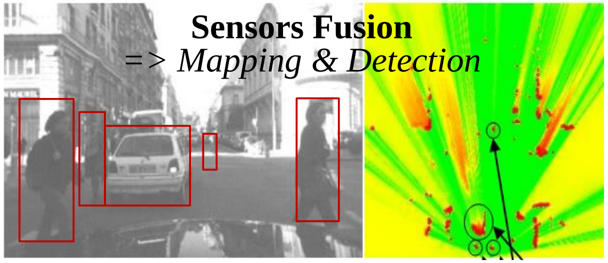
Camera output depends on lighting conditions
Cheap & Rich information & Good for classification

Lidar more accurate & can work at night
Good for fine detection of objects … but ***still Expensive***

- *Any sensor may generate perception errors => It is mandatory to develop* ***Embedded Robust & Efficient "Multi-Sensors Fusion"*** *approaches (e.g. using probabilistic models & filtering)*
- *A new generation of* ***affordable "Solid State Lidars"*** *is supposed to shortly arrive on the market !*
  - *=> No mechanical component & Expected cost less than 1000 US$ before mass production*
  - *=> Numerous announcements since Spring 2016 …* ***but products not yet on the market !!***

Key Technology: Embedded Bayesian Perception

Embedded Multi-Sensors Perception
⇒ Continuous monitoring of the dynamic environment



❑ Main challenges

- ✓ *Noisy data, Incompleteness, Dynamicity, Discrete measurements*
- ✓ ***Strong Embedded & Real time constraints***

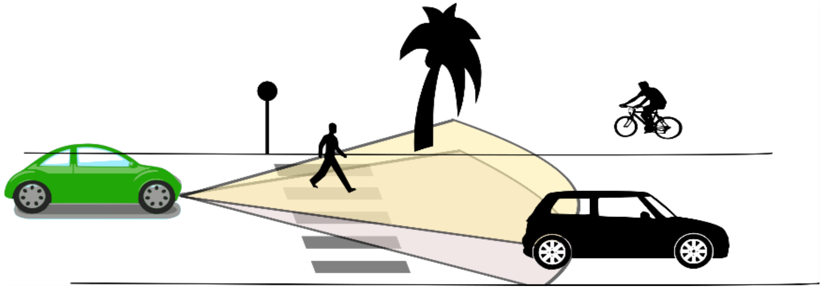
❑ Approach: Embedded Bayesian Perception

- ✓ *Reasoning about **Uncertainty & Time window** (Past & Future events)*
- ✓ ***Improving robustness** using **Bayesian Sensors Fusion***
- ✓ ***Interpreting the dynamic scene** using **Contextual & Semantic** information*
- ✓ ***Software & Hardware integration** using GPU, Multicore, Microcontrollers…*

Bayesian Perception : Basic idea

❑ Multi-Sensors Observations

Lidar, Radar, Stereo camera, IMU …

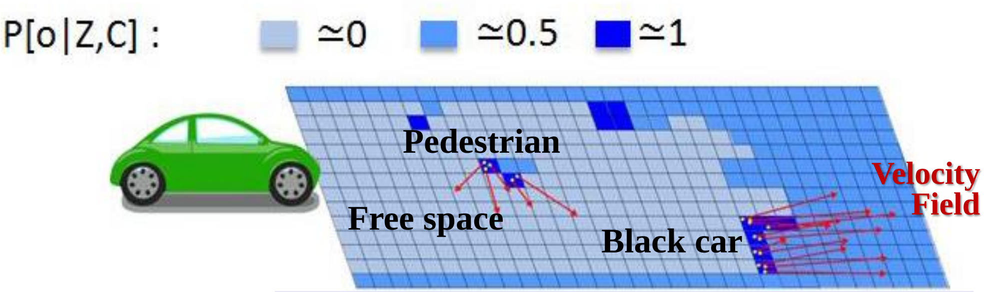


Bayesian
Multi-Sensors Fusion

Real-time

❑ Probabilistic Environment Model

- ✓ *Sensor Fusion*
- ✓ *Occupancy grid integrating uncertainty*
- ✓ *Probabilistic representation of Velocities*
- ✓ *Prediction models*



Concept of Dynamic Probabilistic Grid
⇒ Occupancy & Velocity probabilities
⇒ Embedded models for <u>Motion Prediction</u>

❑ Main philosophy

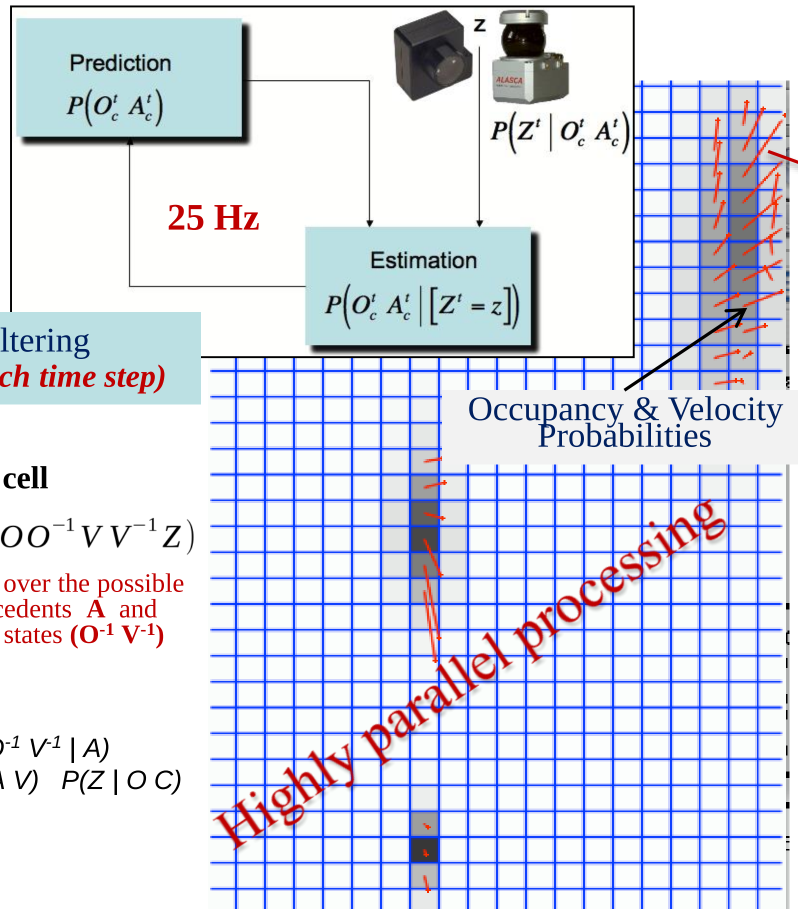
Reasoning at the grid level as far as possible for both :

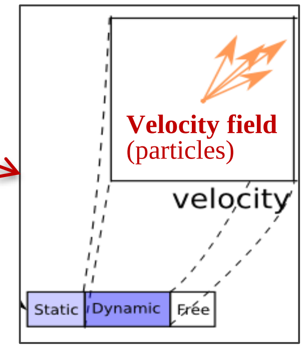
- ***Improving efficiency*** *(highly parallel processing)*
- ***Avoiding traditional object level processing problems*** *(e.g. detection errors, wrong data association…)*

A new framework: Dynamic Probabilistic Grids
=> *A clear distinction between Static & Dynamic & Free components*

[Coué & Laugier IJRR 05] [Laugier et al ITSM 2011] [Laugier, Vasquez, Martinelli Mooc uTOP 2015]

Sensing *(Observations)*





Bayesian Filtering
(Grid update at each time step)

Solving for each cell

$$P(O\,V \mid Z\,C) = \lambda \sum_{A\,O^{-1}\,V^{-1}} P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z)$$

Sum over the possible antecedents **A** and their states **(O^{-1} V^{-1})**

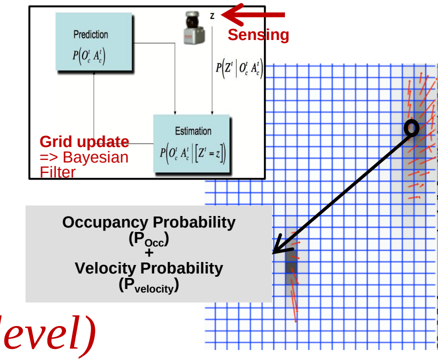
Joint Probability decomposition:

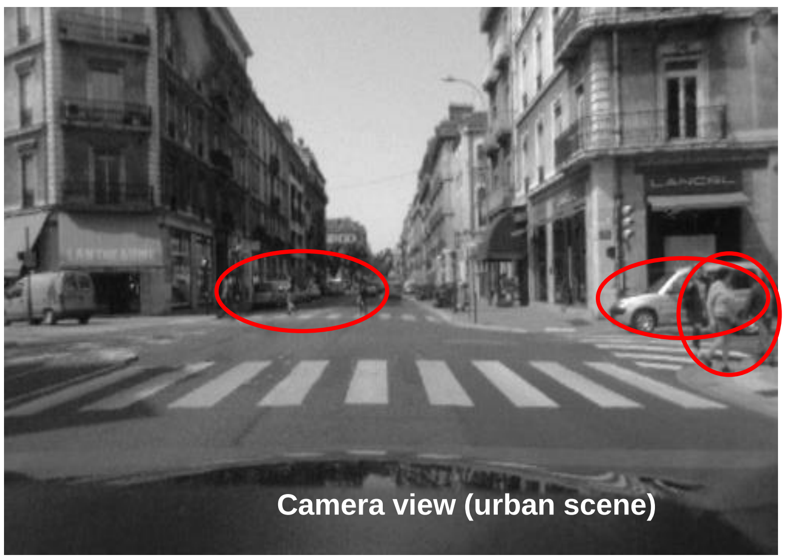
$$P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z) = P(A)\ \ P(O^{-1}\,V^{-1} \mid A)\ \ P(O\,V \mid O^{-1}\,V^{-1})\ \ P(C \mid A\,V)\ \ P(Z \mid O\,C)$$

- ***Bayesian Occupancy Filter (BOF)***
  => Patented by Inria & Probayes
  => Commercialized by Probayes
  => Robust to sensing errors & occultation
- ***Used by:*** *Toyota, Denso, Probayes, Renault, EasyMile, IRT Nanoelec / CEA*
- ***Free academic license available***
- ***Industrial licenses: Toyota, EasyMile***

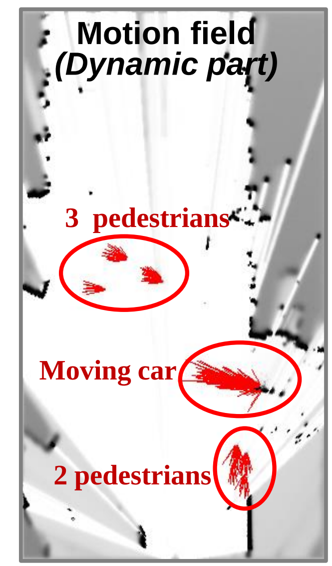
Bayesian Occupancy Filter (BOF) – *Main Features*

- Estimate **Spatial occupancy for each cell of the grid** ***P (O | Z)***
- **Grid update** is performed in <u>each cell in parallel</u> *(using BOF equations)*
- **Extract Motion Field** *(using Bayesian filtering & Fused Sensor data)*
- **Reason at the Grid level** *(i.e. no object segmentation at this reasoning level)*




Camera view (urban scene)

Sensors data fusion
+
Bayesian Filtering

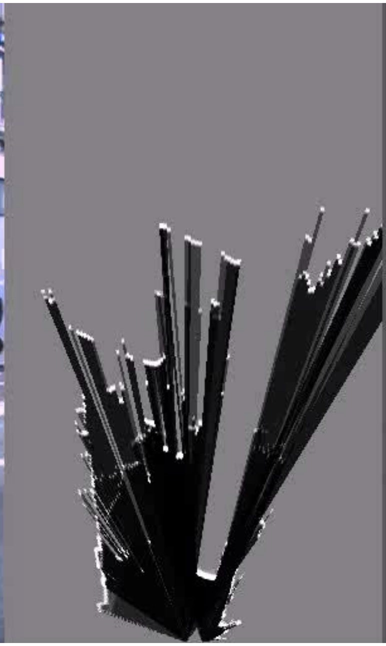


Exploiting the Dynamic information for a better Understanding of the Scene !!

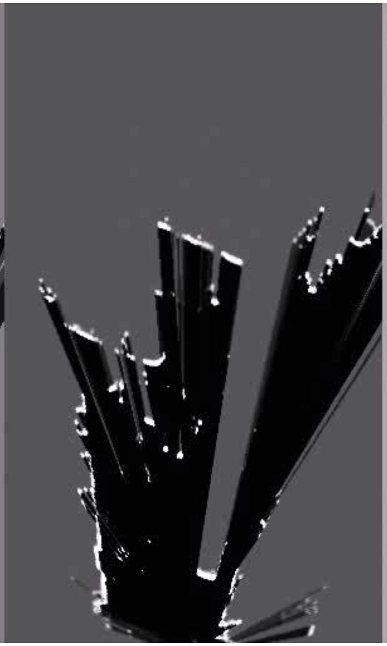
Concept illustration in a dense Urban Environment

Observed Urban Traffic scene

Ego Vehicle *(not visible on the video)*

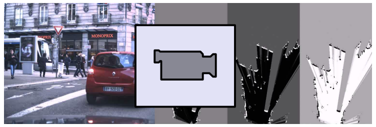


OG Left Lidar



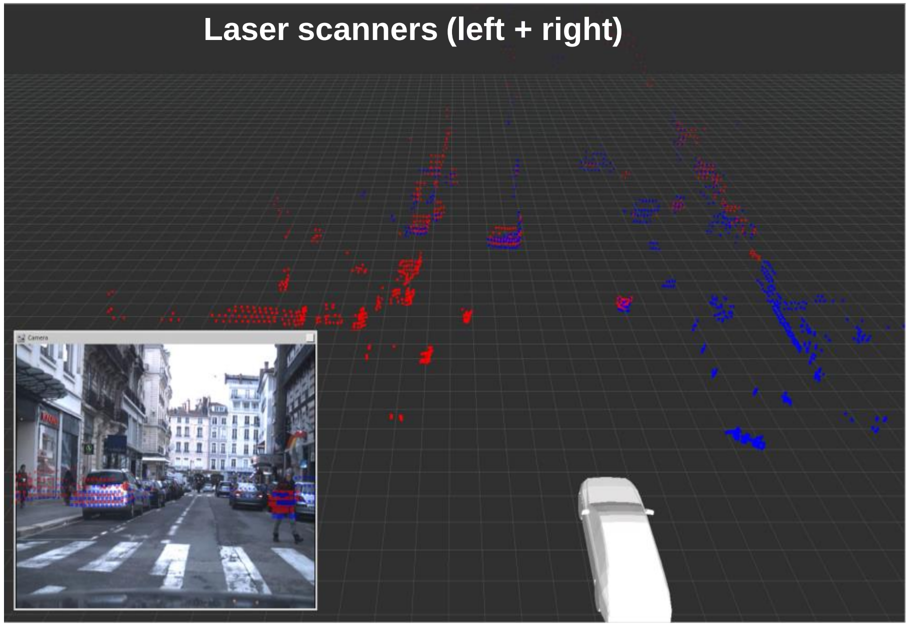
OG Right Lidar

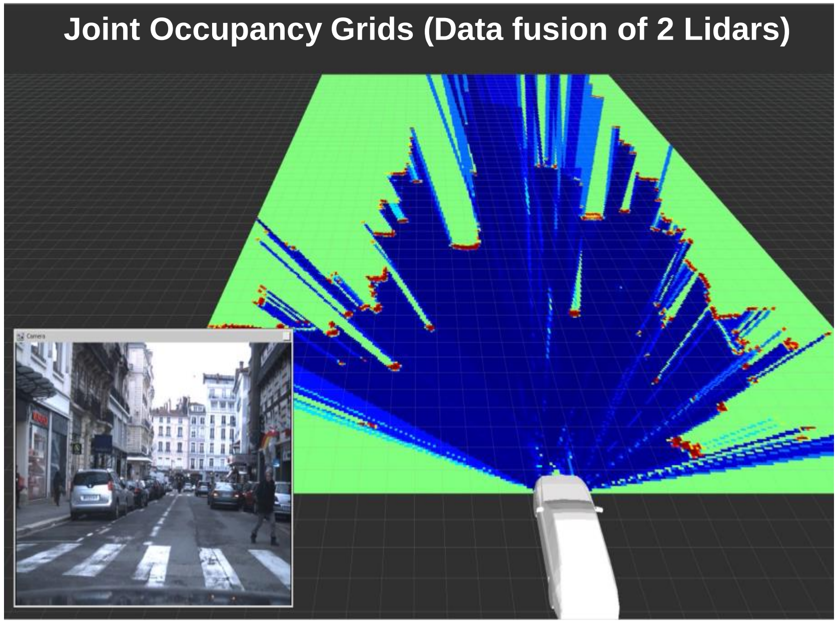
OG Fusion
+
Velocity Fields



Data fusion *–The joint Occupancy Grid*

- **Observations Z_i** are given by each sensor ***i*** *(Lidars, cameras, etc)*
- For each set of observation Z_i , **Occupancy Grids are computed:** $P(O \mid Z_i)$
- Individual grids are **merged into a single one:** $P(O \mid Z)$

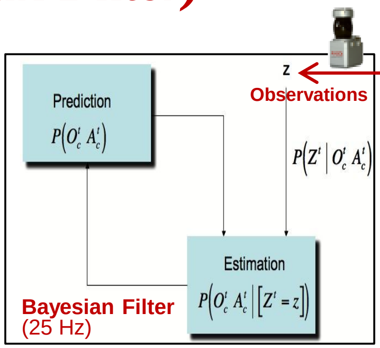




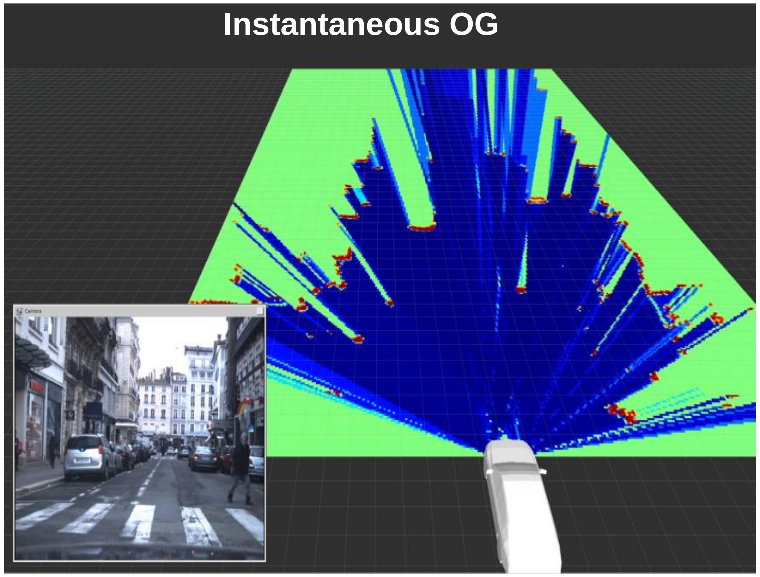
Taking into account dynamicity
=> *Filtered Occupancy Grid (Bayesian filtering)*

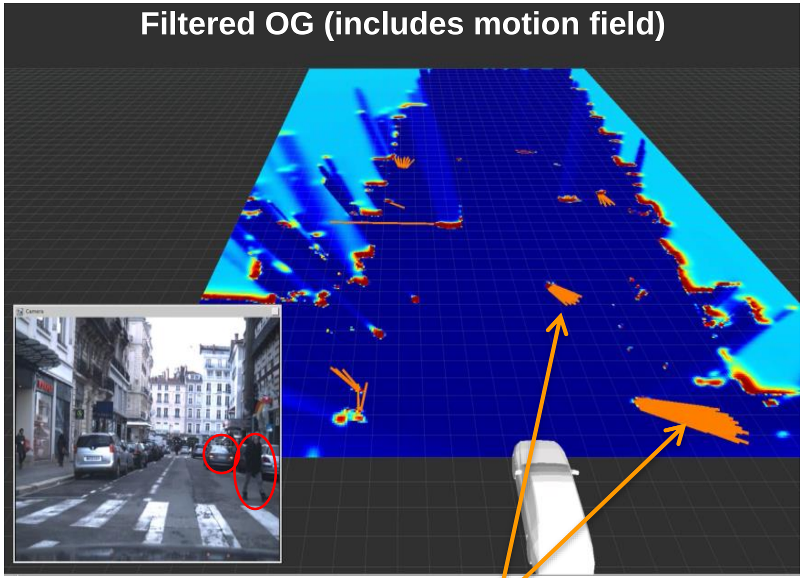
- **Filtering** is achieved through the ***prediction/correction loop*** ***(Bayesian Filter)***

  => *It allows to take into account* ***grid changes over time***



- **Observations** are used to update the environment model
- **Update is performed in each cell <u>in parallel</u>** *(using BOF equations)*
- **Motion field** is constructed from the resulting filtered data

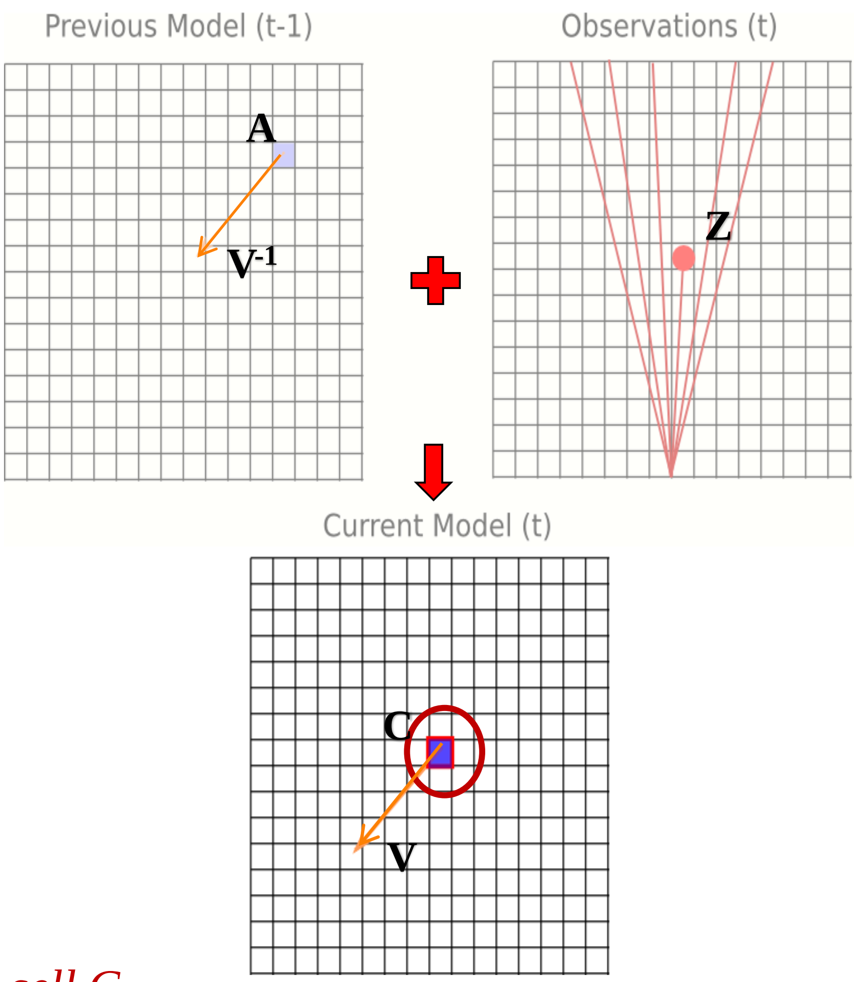




Motion fields are displayed in orange color

Bayesian Occupancy Filter – How it works ?
Formalism

- **Variables**:
  - C : current cell
  - A : antecedent cell, i.e. the cell from which the occupancy of the current cell comes from
  - O : occupancy of the current cell C
  - O^{-1} : previous occupancy in the antecedent cell
  - V : current velocity
  - V^{-1} : previous velocity in the antecedent
  - Z : observations (sensor data)



- **Objective**:
  Evaluate P (O V | Z C)
  *=> Probability of **Occupancy & Velocity** for each cell C,*
  *Knowing the **observations Z** & the **cell location C** in the grid*

Bayesian Occupancy Filter – How it works ?

How to theoretically compute P (O V | Z C) ?

$$P(O\,V\,|\,Z\,C) = \lambda \sum_{A\,O^{-1}\,V^{-1}} P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z)$$

Sum over the possible antecedents **A** and their states **(O^{-1} V^{-1})** at time t-1

The joint probability term can be re-written as follows:

$$P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z) = \boxed{P(A)\quad P(O^{-1}\,V^{-1}\,|\,A)\quad P(O\,V\,|\,O^{-1}\,V^{-1})\quad P(C\,|\,A\,V)\quad P(Z\,|\,O\,C)}$$

Joint probability => *used for the update of P (O V | Z C)*

P(A) *:* Selected as **uniform** *(every cell can a priori be an antecedent)*

$P(O^{-1}\,V^{-1}\,|\,A)$ *:* Result from the previous iteration

$P(O\,V\,|\,O^{-1}\,V^{-1})$ *:* **Dynamic model**

P(C | A V) *:* **Indicator function** of the cell **C** corresponding to the **"projection"** in the grid of the antecedent **A** at a given velocity **V**

P(Z | O C) *:* **Inverse sensor model**

Bayesian Occupancy Filter – How it works ?

How to theoretically compute P (O V | Z C) ?

$$P(O\,V\,|\,Z\,C) = \lambda \sum_{A\,O^{-1}\,V^{-1}} P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z)$$

Sum over the possible antecedents **A** and their states **(O^{-1} V^{-1})** at time t-1

The joint probability term can be re-written as follows:

$$P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z) = P(A)\;\; P(O^{-1}\,V^{-1}\,|\,A)\;\; P(O\,V\,|\,O^{-1}\,V^{-1})\;\; P(C\,|\,A\,V)\;\; P(Z\,|\,O\,C)$$

But, computing this expression is difficult in practice

=> Huge range of possible antecedents

=> Strongly depends on Grid size & Velocity range

Step 1: Occupancy Grid Construction
The Sensor Model

- **Sensors data**
  - Incomplete (objects states are only partially measurable)
  - Uncertain (measures are noisy)

- **The sensor model**

  => Modeling the relationship between **Objects** **true states** and the **corresponding** **Observations** made by sensors

- **Probabilistic representation** (Thrun 2005 [1])

  => Inverse sensor model: $\mathbf{P\ (Z_t \mid S_t)}$

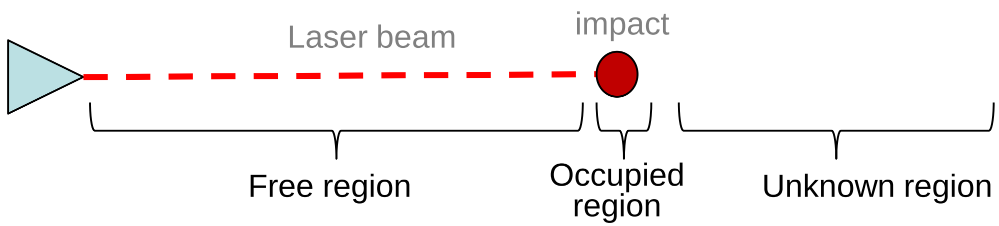
  Observation given by the sensor (Z_t)

  Object is at state S_t

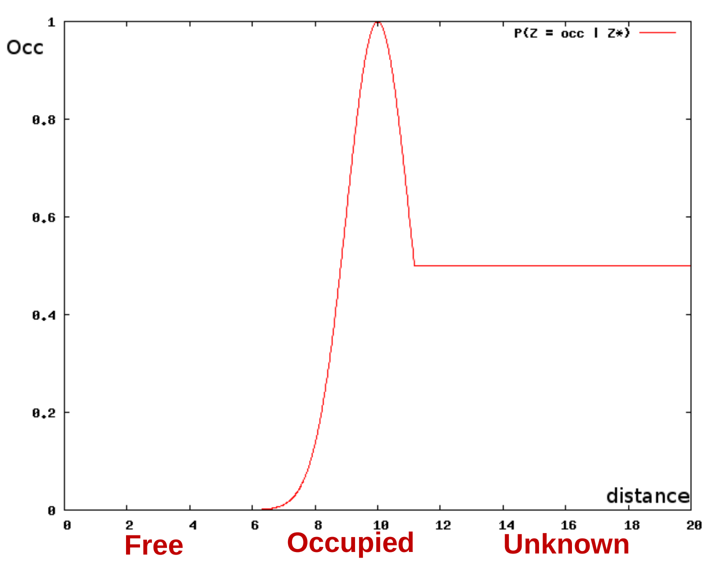
[1] Probabilistic Robotics, Thrun 2005

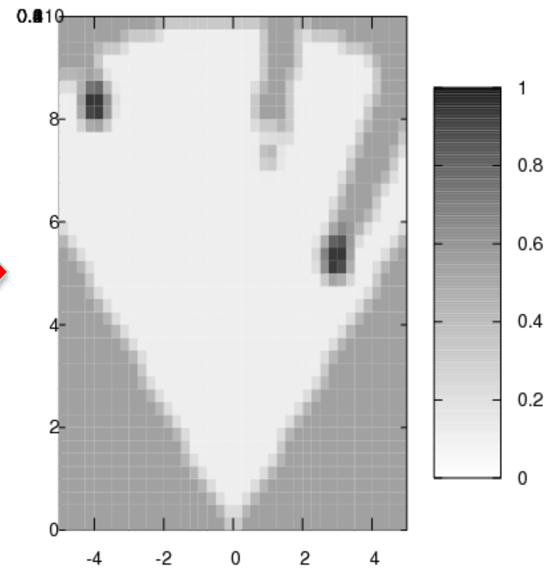
Step 1: OG Construction – *The Lidar Sensor Model*

- **Basic idea:** A Laser beam split the space in 3 regions (Free, Occupied, unknown)



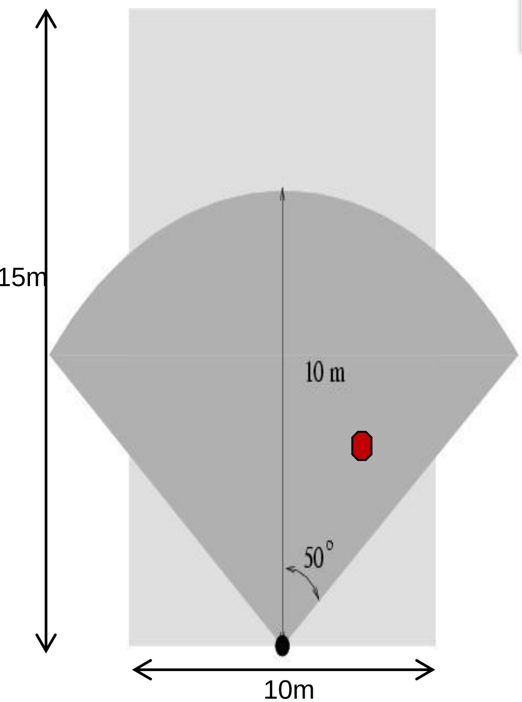
- **Probabilistic modeling:** The independent beam model



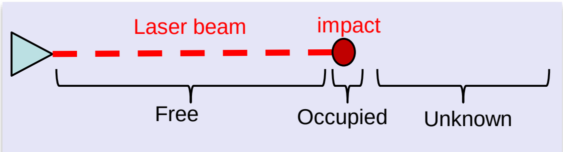


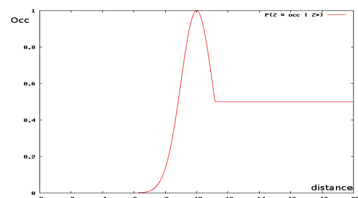
[1] Probabilistic Robotics, Thrun 2005

Step 1: OG Construction – *Example 1*



1 Sensor (laser scanner)
1 Stationary object => *Observation z*

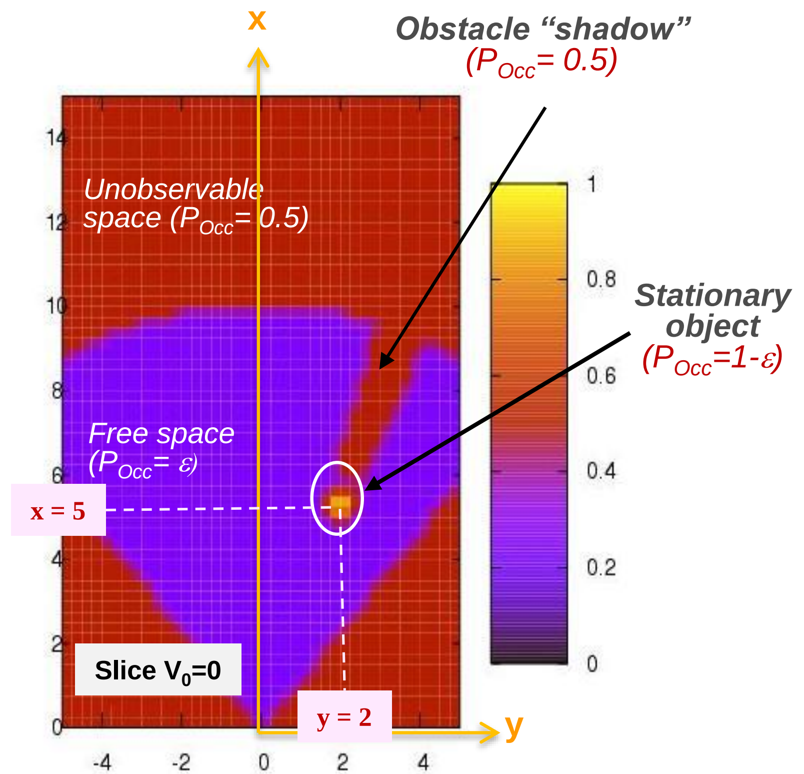




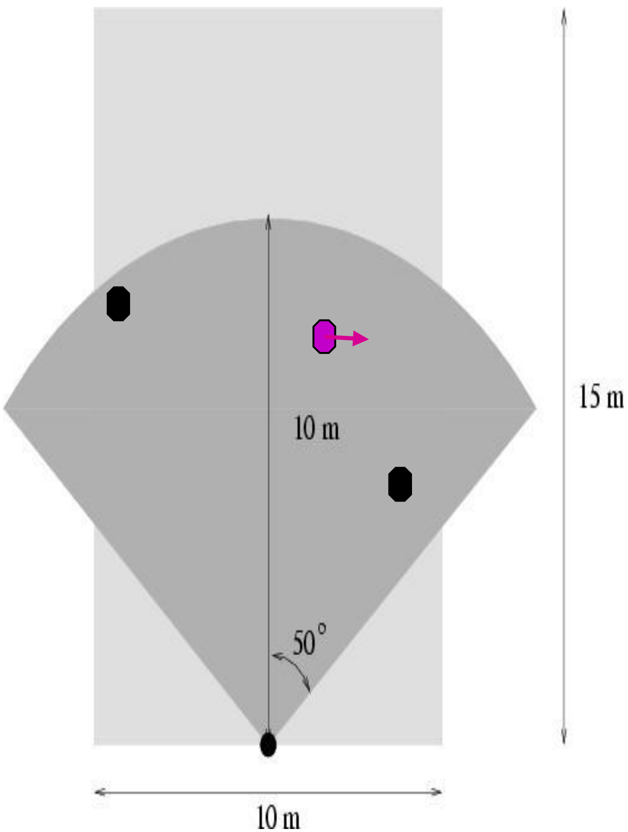
Independent beam model

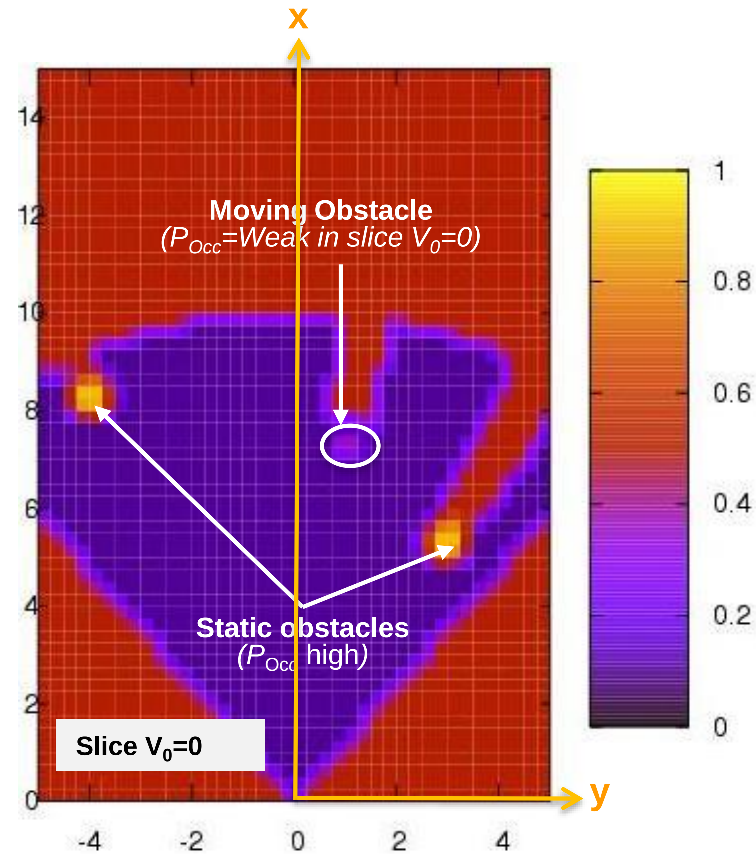
Sensor observation: z = (5, 2, 0, 0)
Cell characterization: c = [x, y, 0, 0]
Position   Velocity (v_x, v_y)

$P_{Occ} = P([O_c = occ] \mid z\ c)$



Step 1: OG Construction – *Example 2*



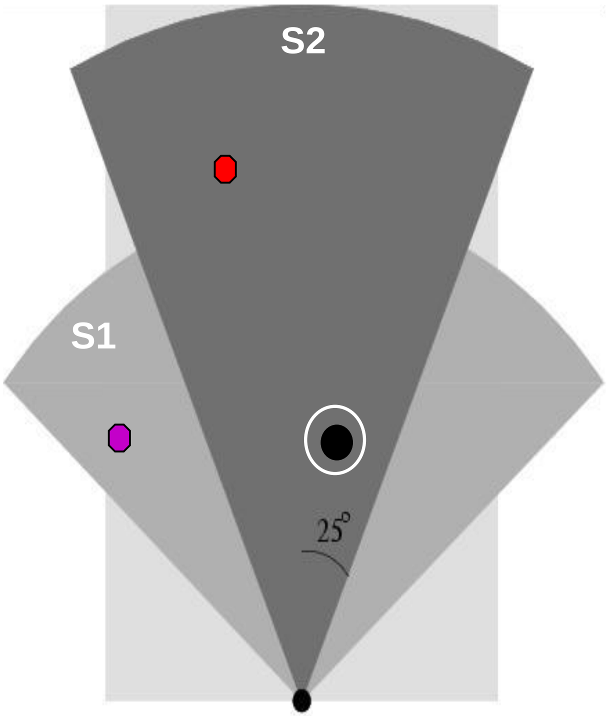


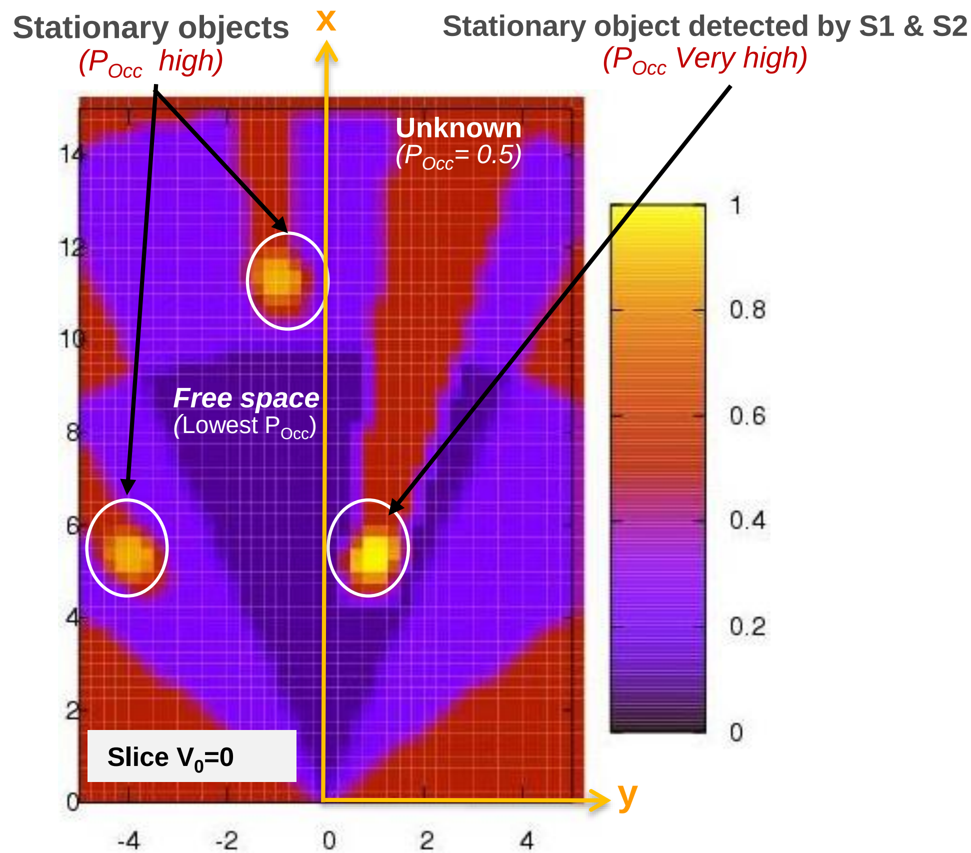
1 Sensor
2 Stationary objects => *Observations (z_1, z_3)*
1 Moving object => *Observation z_2*

$$P_{Occ} = P([O_c=occ] \mid z_1\, z_2\, z_3\;\; c)$$

z_1 = (8.3, -4, 0, 0)
z_2 = (7.3, 1.9, 0, 0.8)
z_3 = (5, 3, 0, 0)
c = [x, y, 0, 0] ***=> in velocity slice V_0=0***

Step 1: OG Construction – *Example 3*





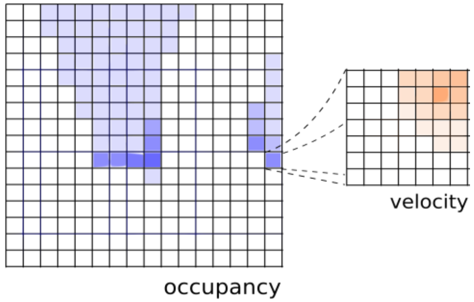
2 Sensors (S1 & S2)
3 Stationary objects
⇒ *Observations sensor S1:* $Z_{1,1}$, $Z_{1,2}$
⇒ *Observations sensor S2:* $Z_{2,1}$, $Z_{2,2}$
⇒ *Black object detected by S1 & S2*

$$P_{Occ} = P([O_c = occ] \mid z_{1,1}\ z_{1,2}\ z_{2,1}\ z_{2,2}\ c)$$

$z_{1,1}$ = (5.5, -4, 0, 0) $z_{1,2}$ = (5.5, 1, 0, 0)
$z_{2,1}$ = (11, -1, 0, 0) $z_{2,2}$ = (5.4, 1.1, 0,0)
c = [x, y, 0, 0]

Step 2: How to compute P(OV | Z C) in practice?
Initial approach: The classic BOF filtering process

- **Initial implementation:**
  - ✓**Regular grid**
  - ✓**Transition histograms** for every cell (for representing velocities)

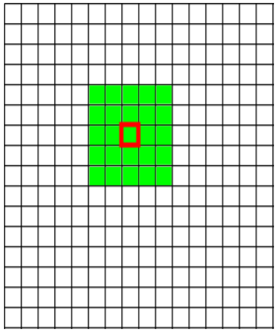


- **Practical computation:**

$$P(O\,V\,|\,Z\,C) = \lambda \sum_{A\,O^{-1}\,V^{-1}} P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z)$$

Sum over the possible antecedents A and their states (O^{-1} V^{-1})

=> Sum over the **neighborhood,** with a **single possible velocity per antecedent A** of equation:

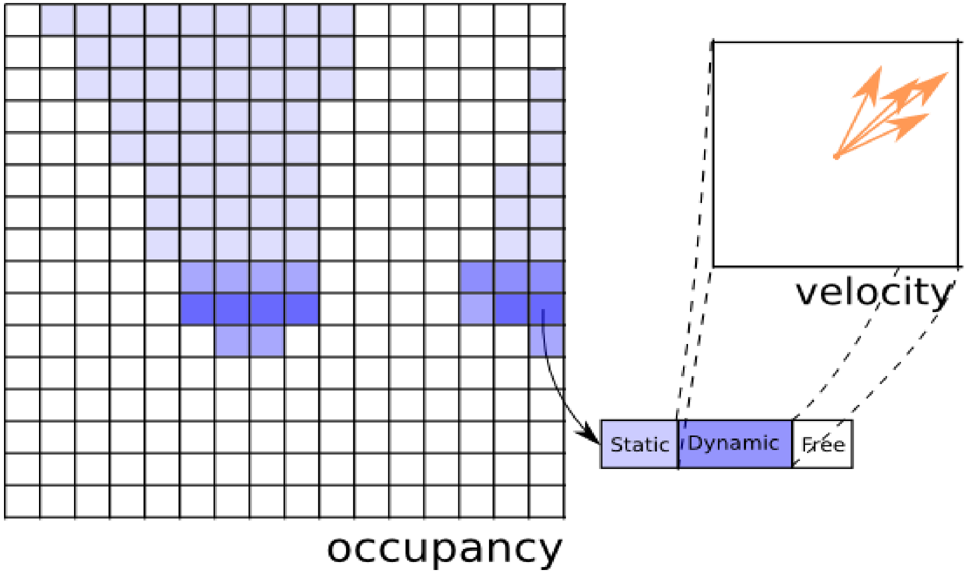


$$P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z) = P(A)\ \ P(O^{-1}\,V^{-1}\,|\,A)\ \ P(O\,V\,|\,O^{-1}\,V^{-1})\ \ P(C\,|\,A\,V)\ \ P(Z\,|\,O\,C)$$

- **Drawbacks:**
  - ⇒ Large memory size required (velocity histograms large & almost empty)
  - ⇒ Weak accuracy
  - ⇒ Temporal & Spatial Aliasing problems

Step 2: How to compute P(OV | Z C) in practice?
Improved approach: HSBOF updating process (principle)

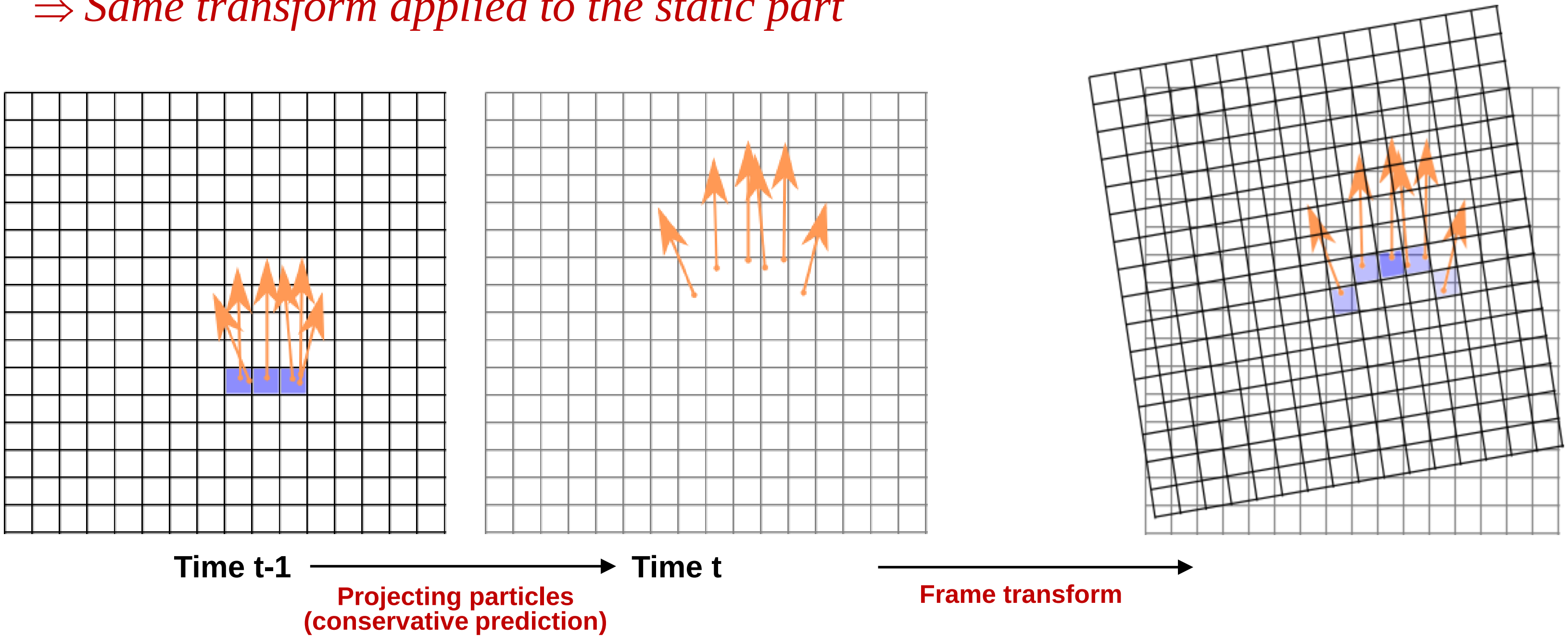
- **Basic idea**: *Modify the representation structure to avoid the previous computational problems*
  - ✓ Making a clear distinction between **Static & Dynamic & Free** components
  - ✓ Modeling velocity using **Particles** *(instead of histogram)*
  - ✓ Making an **adaptive repartition** of those particles in the grid



Step 2: How to compute P(OV | Z C) in practice?
Improved approach: HSBOF updating process (principle)

- **Introducing a Dynamic model for "projecting" particles in the grid** *($S_{t-1} \rightarrow S_t$)*
  - ⇒ *Immediate antecedent association*
  - ⇒ *Simplified velocity prediction to the cells*

- **Updating Grid Reference Frame (the car & sensors are in motion)**
  - ⇒ *Translation & Rotation values **provided by sensors** (Odometry + IMU)*
  - ⇒ *Same transform applied to the static part*



Step 2: How to compute P(OV | Z C) in practice ?

The HSBOF filtering calculation process

$$P(O\,V\,|\,Z\,C) = \lambda \sum_{A\,O^{-1}\,V^{-1}} P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z)$$

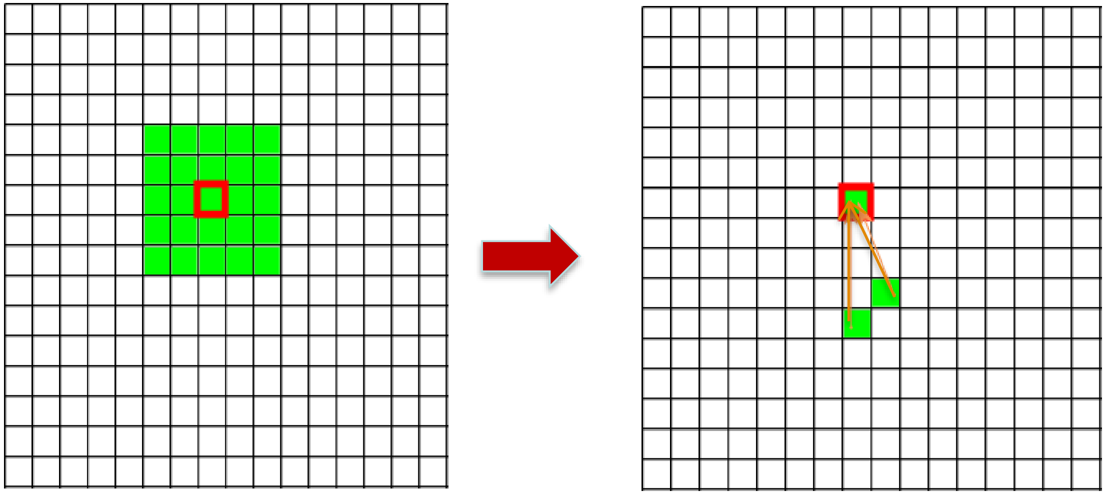
~~Sum over the neighborhood, with a single velocity per antecedent~~

A more efficient computation approach :

=> *Sum over* ***the particles projected in the cell*** *&* *their* ***related static parts***

$$P(C\,A\,O\,O^{-1}\,V\,V^{-1}\,Z) = P(A)\;\; P(O^{-1}\,V^{-1}\,|\,A)\;\; P(O\,V\,|\,O^{-1}\,V^{-1})\;\; P(C\,|\,A\,V)\;\; P(Z\,|\,O\,C)$$

Previous computation approach (histograms)



New computation approach (particles)

Step 2: How to compute P(OV | Z C) in practice?
HSBOF updating process (outline of the algorithm)

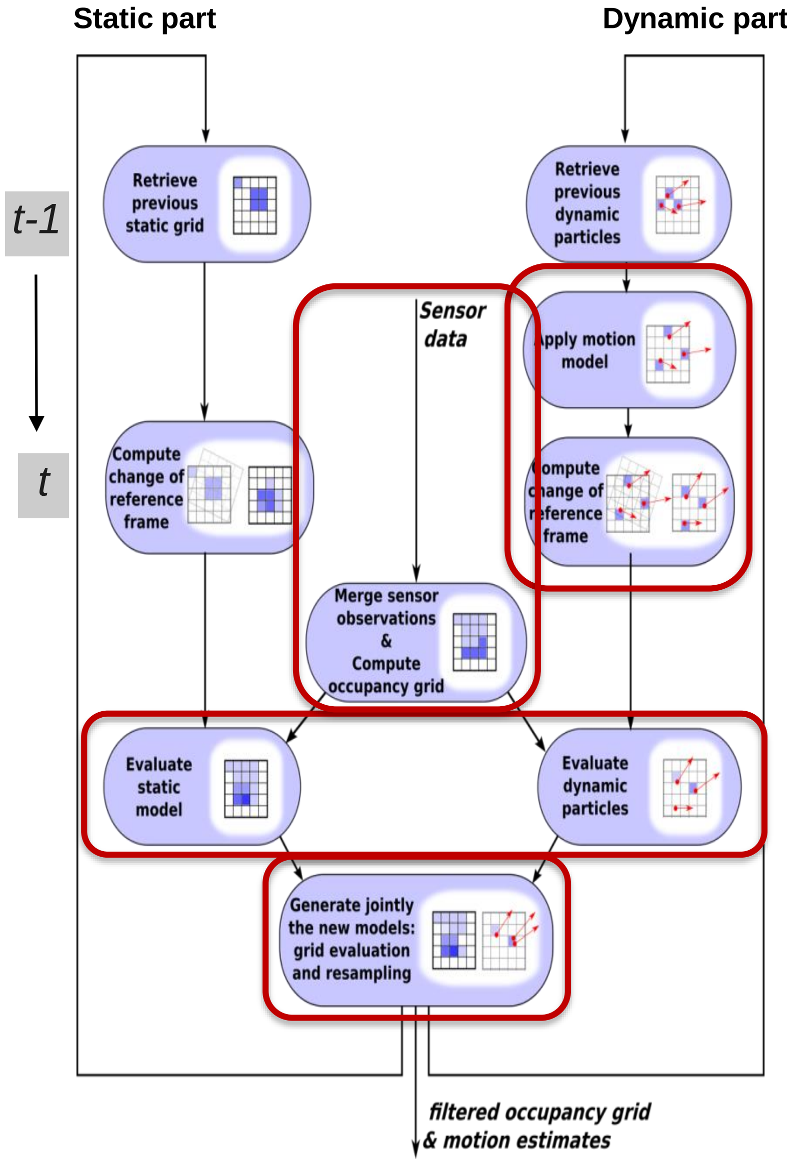
Main steps in the updating process

• Dynamic part (particles) is "**projected**" in the grid using motion model => *motion prediction*

• Both Dynamic & Static parts are expressed in the **new reference frame** => *moving vehicle frame*

• The two resulting representations are confronted to the **observations** => *estimation step*

• **New representations (static & dynamic)** are jointly evaluated and particles re-sampled



Content of the Tutorial

- ❑ Socio-economic & Technological Context + State of the Art
- ❑ Decisional & Control Architecture – Outline
- ❑ Bayesian Perception *(key Technology 1)*
- ❑ **Embedded Bayesian Perception & Experimental results**
- ❑ Bayesian Risk Assessment & Decision-making *(Key Technology 2*

Recent implementations & Improvements

*Several implementations (models & algorithms) more and more adapted to **Embedded constraints & Scene complexity***

- ❖ Hybrid Sampling Bayesian Occupancy Filter (HSBOF, 2014) *[Negre et al 14] [Rummelhard et al 14]*

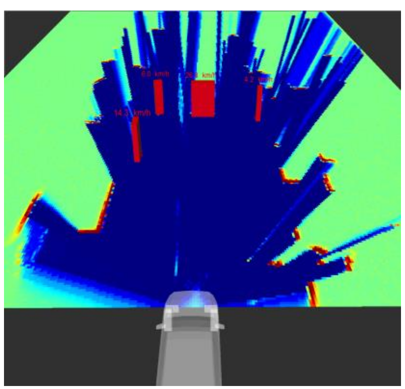
  *=> Drastic **memory size reduction** (factor 100) + Increased **efficiency** (complex scenes)*
  *+ More **accurate Velocity estimation** (using Particles & Motion data from ego-vehicle)*

- ❖ Conditional Monte-Carlo Dense Occupancy Tracker (CMCDOT, 2015) *[Rummelhard et al 15]*

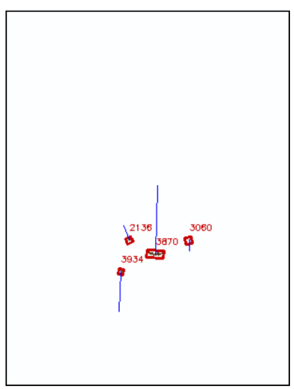
  ***=> Increased efficiency using** "state data" (Static, Dynamic, Empty, **Unknown**) + Integration of a **"Dense Occupancy Tracker"** (Object level, Using particles propagation & ID)*

- ❖ CMCDOT + Ground Estimator (Patent 2017) *[Rummelhard et al 17]*

  => Ground shape estimation & Improve obstacle detection (avoid false detections on the ground)



Grid & Pseudo-objects



Tracked Objects

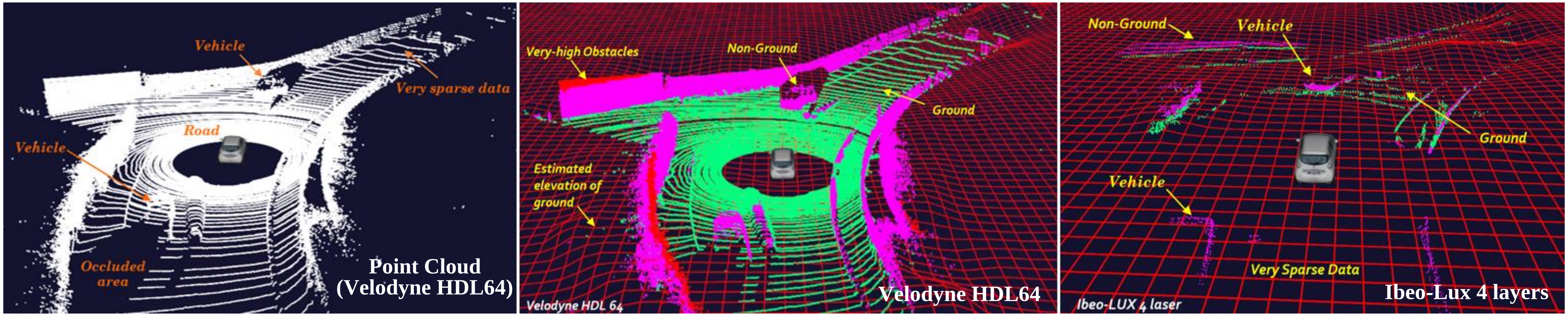
Classification *(using Deep Learning)*

Detection & Tracking & Classification

Ground Estimation & Point Cloud Classification

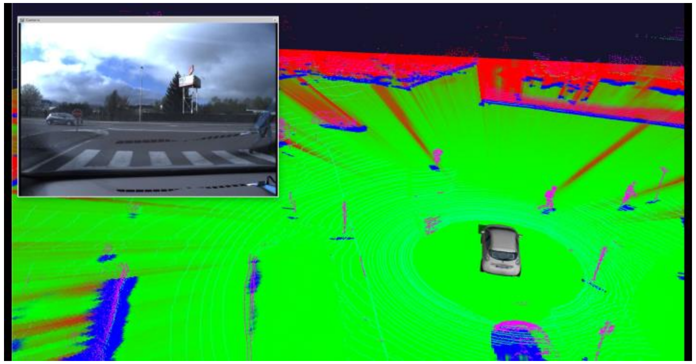
⇒ *Smart OG generation taking into account the* ***ground shape*** *&* ***height of laser impacts***

⇒ *Process properly sensors data (for OG & DATMO) & Avoid false detections on the road surface*



•***Ground estimation :*** *1m x 1m ground node resolution*
=> *ground-points in green*, *obstacles in pink / red*

•***Occupancy grid :*** *Images 0.1m x 0.1m occupancy grid resolution*
=> *green for free space*, *blue occupied*, *red unknown*



❖ Ground model constructed using a *Spatio-Temporal Conditional Random Field*, estimated through efficient parallelized process [1]

❖ Model accurate enough to represent rolling roads & Efficient enough for **real-time** performances on embedded devices

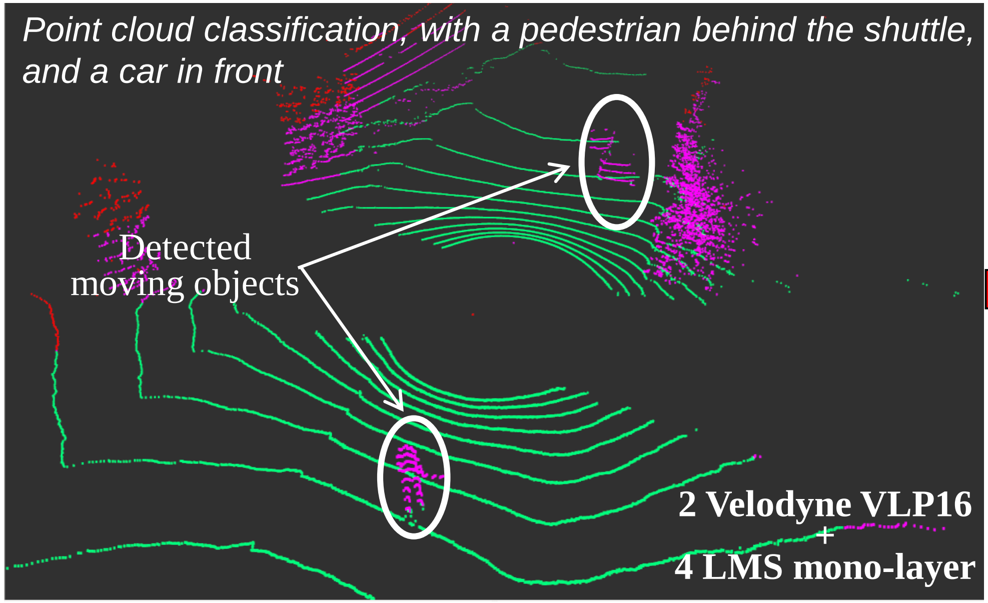
❖ The complete system (including CMCDOT) has been implemented on a Nvidia Tegra X1

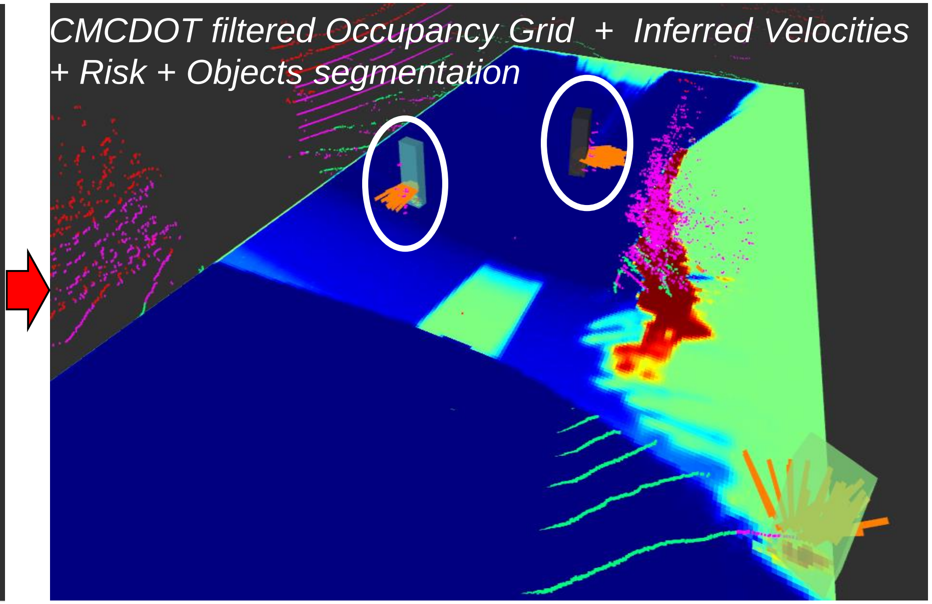
[1] Ground estimation and point cloud segmentation using spatio-temporal conditional random field, Rummelhard et al, IV 2017, Redondo Beach, June 2017

Integration on a commercial vehicle

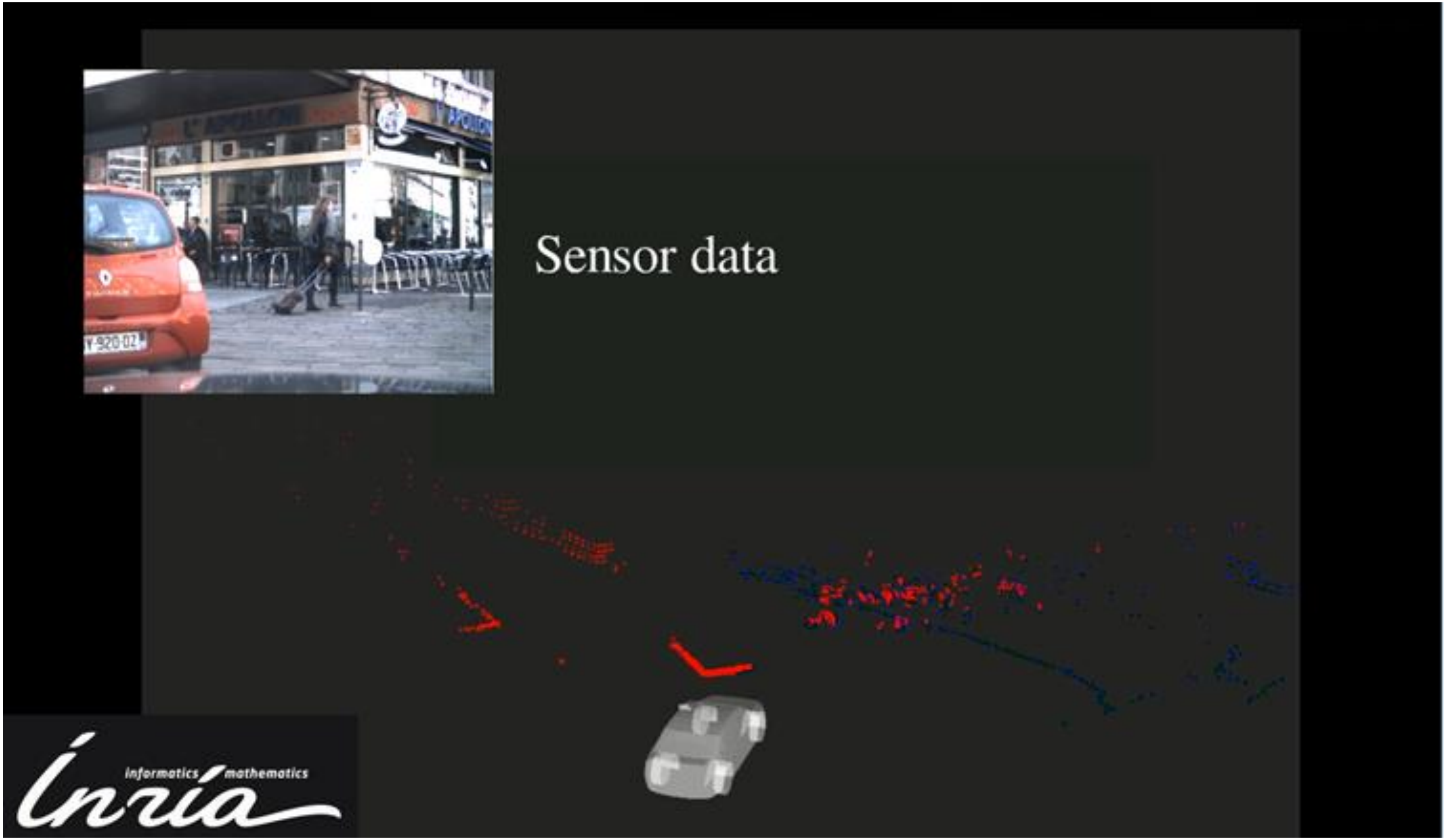
- **POC 2017: Complete system implemented on Nvidia TX1**, and easily connected to the shuttle system network *in a few days* (using ROS)

- **Shuttle sensor data** has been fused and processed in **real-time**, with a successful Detection & Characterization of the **Moving & Static Obstacles**
- **Full integration on a commercial product** under development with an industrial company (confidential)





CMCDOT – *Complete process illustration*

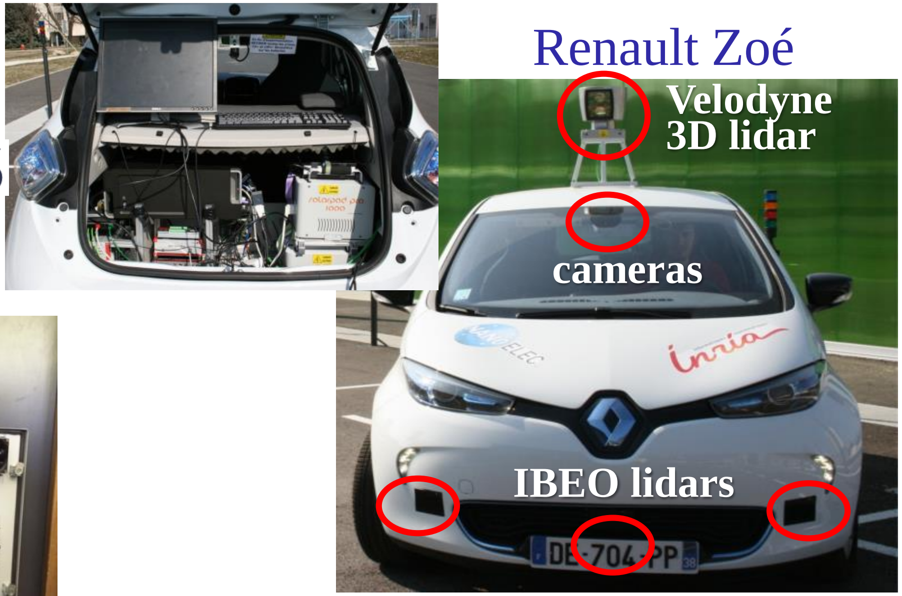


Experimental Vehicles & Connected Perception Units

Toyota Lexus

ROS

Renault Zoé



Connected Perception Unit

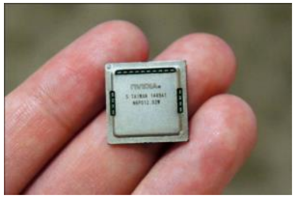
=> Same embedded perception systems than in vehicles

Nvidia GTX Titan X
Generation Maxwell

Nvidia GTX Jetson TK1
Generation Maxwell

Nvidia GTX Jetson TX1
Generation Maxwell

Software / Hardware Integration *– GPU implementation*



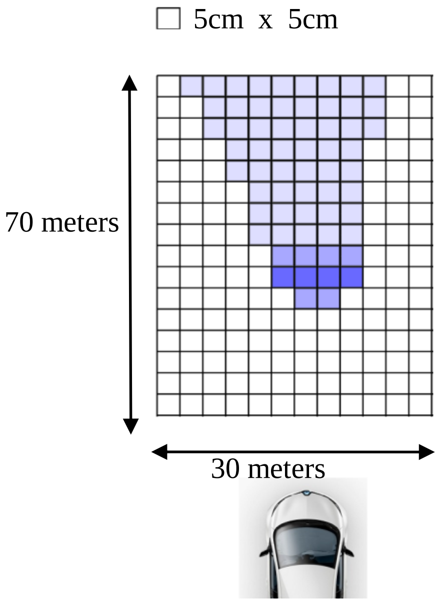
- Highly parallelizable framework, **27 kernels** over cells and particles
  => Occupancy, speed estimation, re-sampling, sorting, prediction
- Real-time implementation (20 Hz), optimized using Nvidia profiling tools

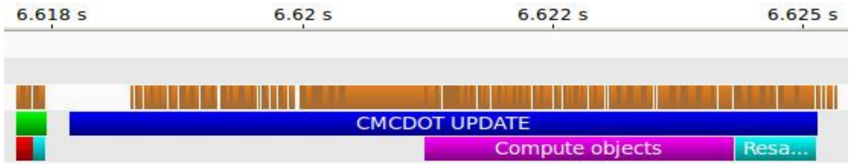
Results:

- Configuration with 8 Lidar layers (2x4)
- Grid: 1400 x 600 (840 000 cells) **+** Velocity samples: 65 536


=> Jetson TK1: *Grid Fusion **17ms***, *CMCDOT **70ms***


=> Jetson TX1: *Grid Fusion **0.7ms***, *CMCDOT **17ms***

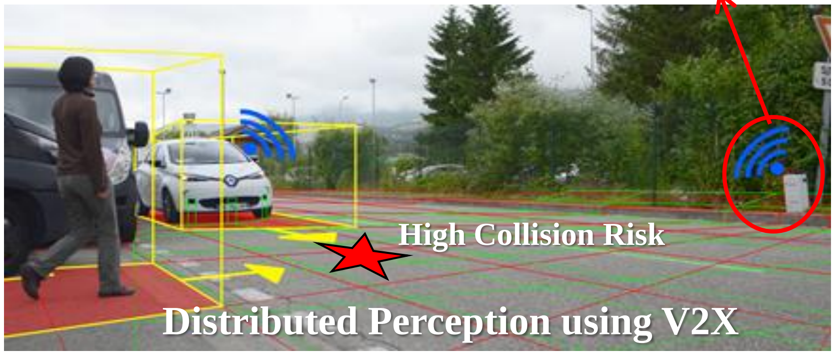




Experimental Areas

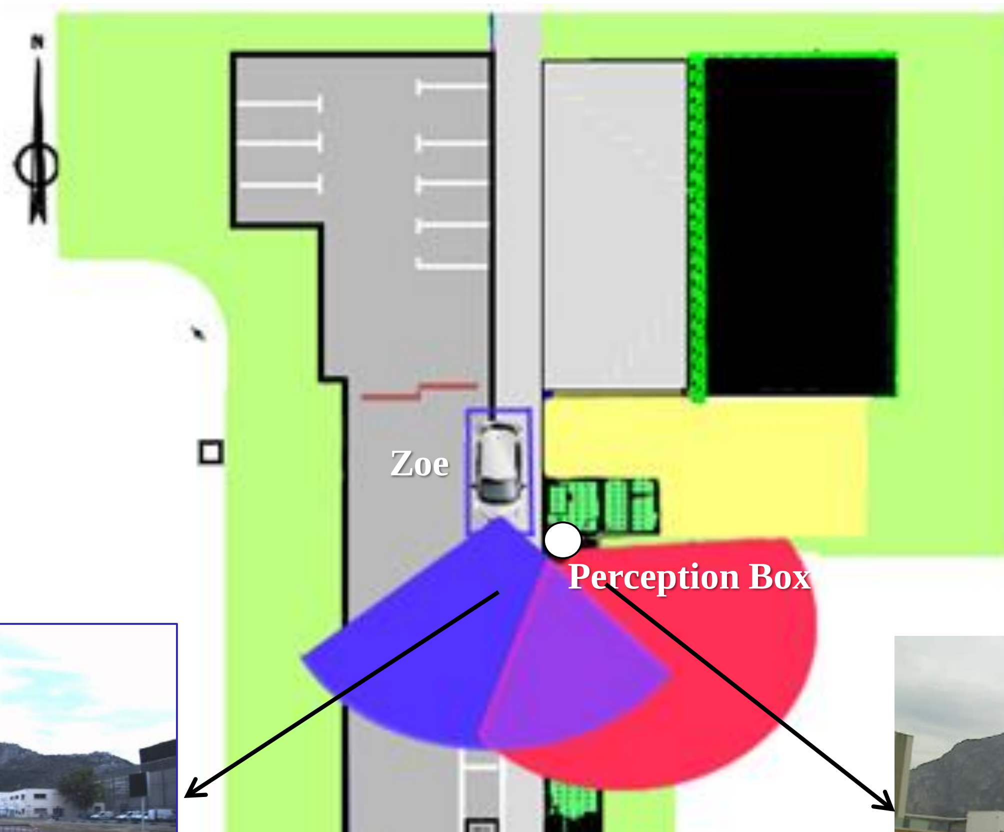
Connected
Perception Unit

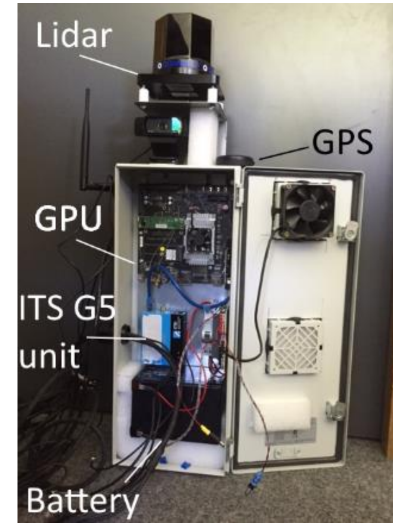
- **Protected experimental area**



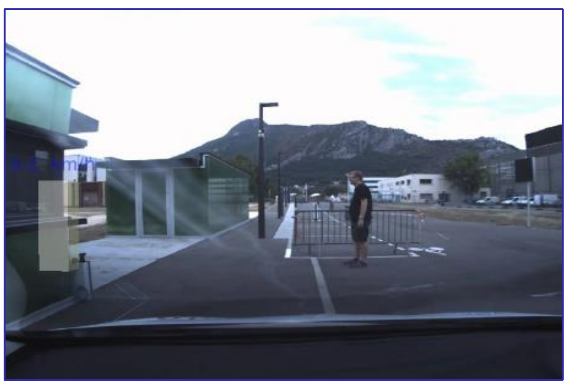
- **Open real traffic (Urban & Highway)**

V2X: Distributed Perception experiment using CMCDOT

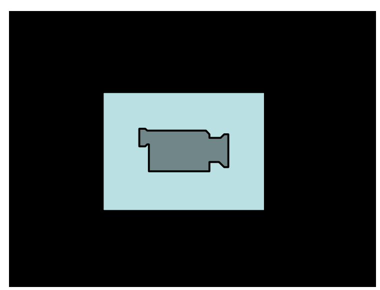


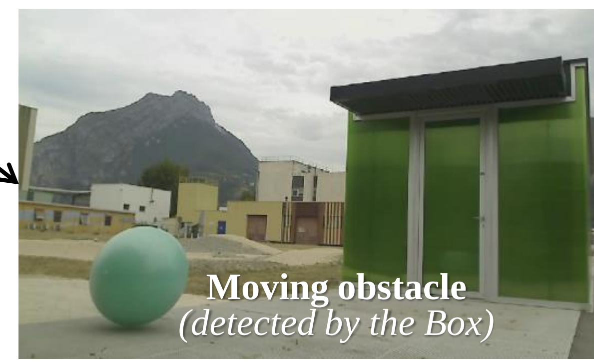


Connected
Perception Unit



Camera Image provided by the
Zoe vehicle





Camera Image provided by the
Perception box

V2X: Data exchange & Synchronization

❑ Data exchange

ROS

ITRI

GPS position + Velocity + Bounding box
(broadcast)

Collision Risk (CMCDOT)
(space & time localization, probability)

ITRI

ITS-G5 (Standard ITS Geonetworking devices)
Basic Transport Protocal IEEE 802.11p

❑ Synchronization

Chrony (**N**etwork **T**ime **P**rotocol)

GPS Garmin + PPS Signal (1 pulse per second)

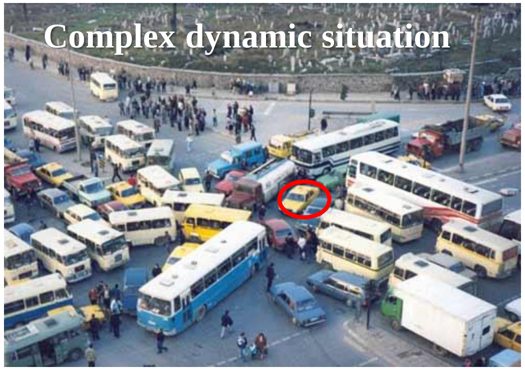
Serial Port

GPIO + UART

Content of the Tutorial

- ❑ Socio-economic & Technological Context + State of the Art
- ❑ Decisional & Control Architecture – Outline => Not presented
- ❑ Bayesian Perception *(key Technology 1)*
- ❑ Embedded Bayesian Perception & Experimental results
- ❑ **Bayesian Risk Assessment & Decision-making *(Key Techno 2)***

Key Technology 2: Risk Assessment & Decision
=> Decision-making for avoiding Pending & Future Collisions



❑ Main challenges

Uncertainty, Partial Knowledge, World changes, ***Human in the loop + Real time***

❑ Approach: Prediction + Risk Assessment + Bayesian Decision-making

- ✓ *Reason about* ***Uncertainty & Contextual Knowledge*** *(using* ***History*** *&* ***Prediction****)*
- ✓ ***Estimate probabilistic Collision Risk at a given time horizon*** $t+\delta$
- ✓ ***Make Driving Decisions*** *by taking into account the* ***Predicted behavior*** *of all the observed surrounding traffic participants (cars, cycles, pedestrians …) &* ***Social / Traffic rules***

Underlying Conservative Prediction Capability

=> *Application to Conservative Collision Anticipation*

[Coué & Laugier IJRR 05]

Autonomous Vehicle (Cycab)

Parked Vehicle (occultation)

Pioneer Results (2005)

Thanks to the prediction capability of the BOF technology, the Autonomous Vehicle "anticipates" the behavior of the pedestrian and brakes *(even if the pedestrian is temporarily hidden by the parked vehicle)*

Step 1: Short-term collision risk – *Main features*

=> Grid level & Conservative motion hypotheses (proximity perception)

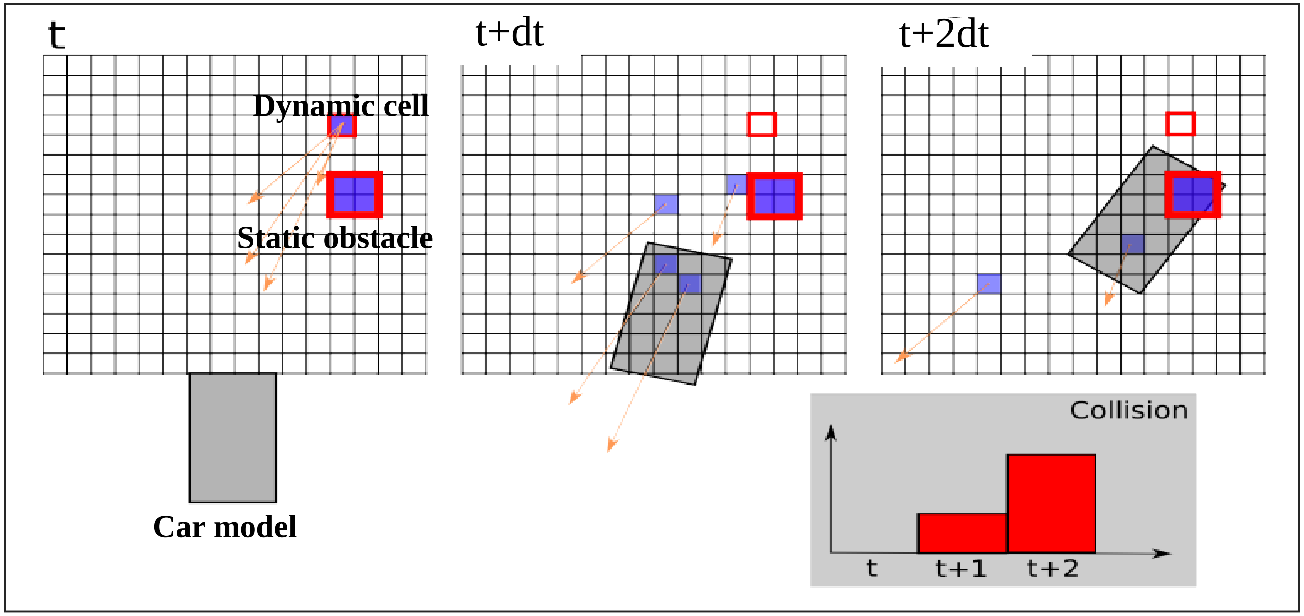
Proximity perception: d <100m and t <5s
δ= 0.5 s => Precrash
δ= 1 s => Collision mitigation
δ > 1.5s => Warning / Emergency Braking

Main Features

- *Detect **"Risky Situations"** a few seconds ahead (3-5s)*
- *Risky situations are **both localized in Space & Time***
  - ⇒ ***Conservative Motion Prediction*** *in the grid (Particles & Occupancy)*
  - ⇒ ***Collision checking*** *with* ***Car model*** *(shape & velocity) for* ***every future time steps*** *(horizon h)*
- *Resulting information can be used for choosing **Avoidance Maneuvers***

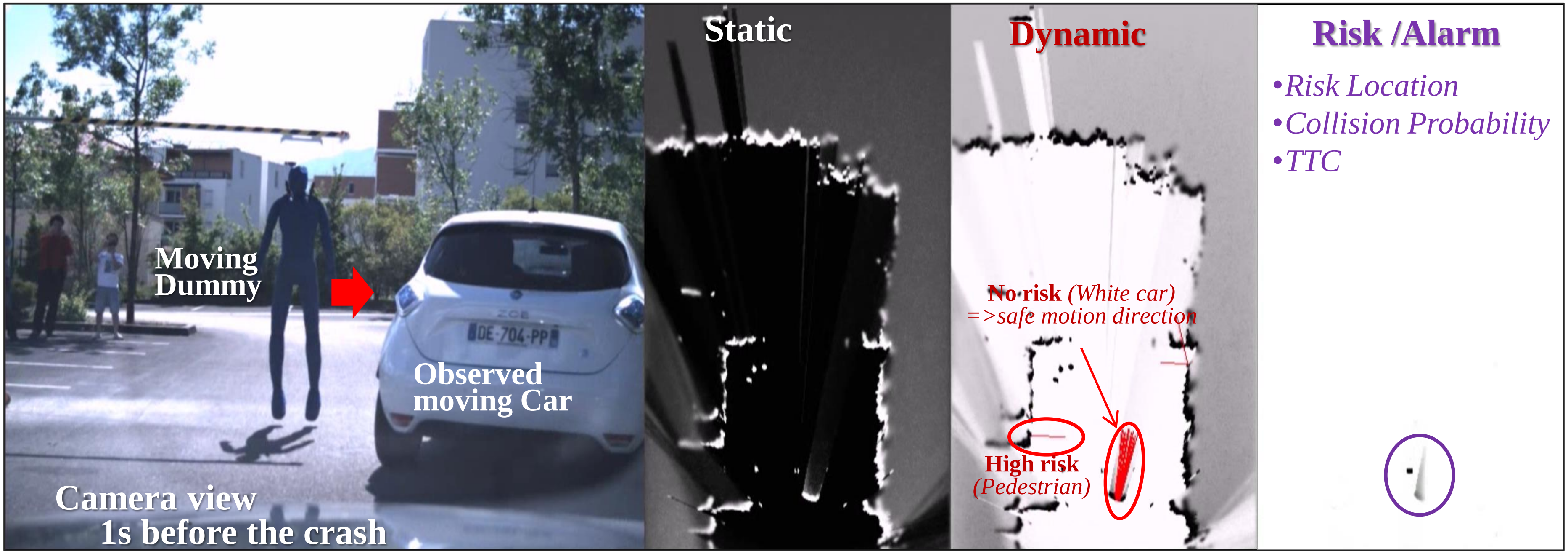
Collision Risk Estimation: *Integration of risk over a time range [t t+δ]*

*=> Projecting over time the **estimated Scene changes** (DP-Grid) & **Car Model** (Shape + Motion)*



Short-term collision risk – *System outputs (real-time)*

=> Static & Dynamic grids + Risk assessment

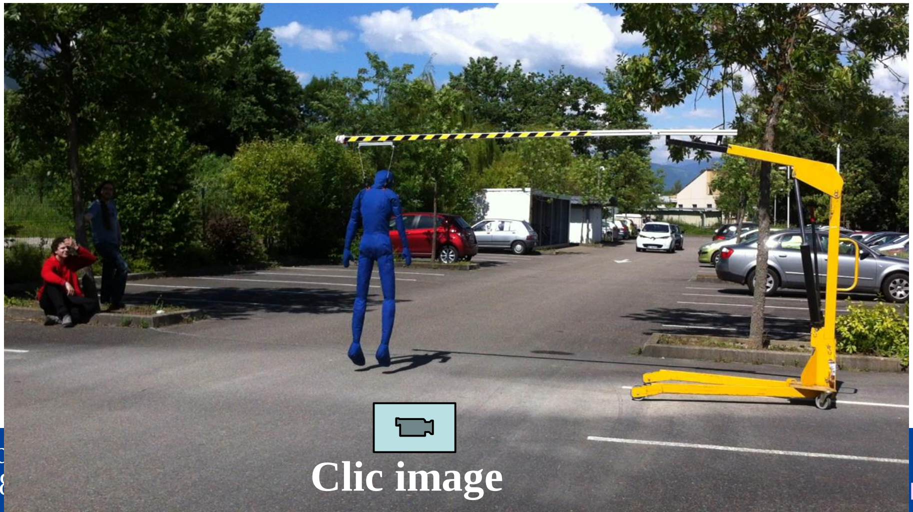


○ **FAQ :** *What happen if some velocities change after that the collision risk for the next 3s has been evaluated ?*

○ **Answer:** *The collision risk is recomputed at the next time step (i.e max 40ms after the change of dynamics).*

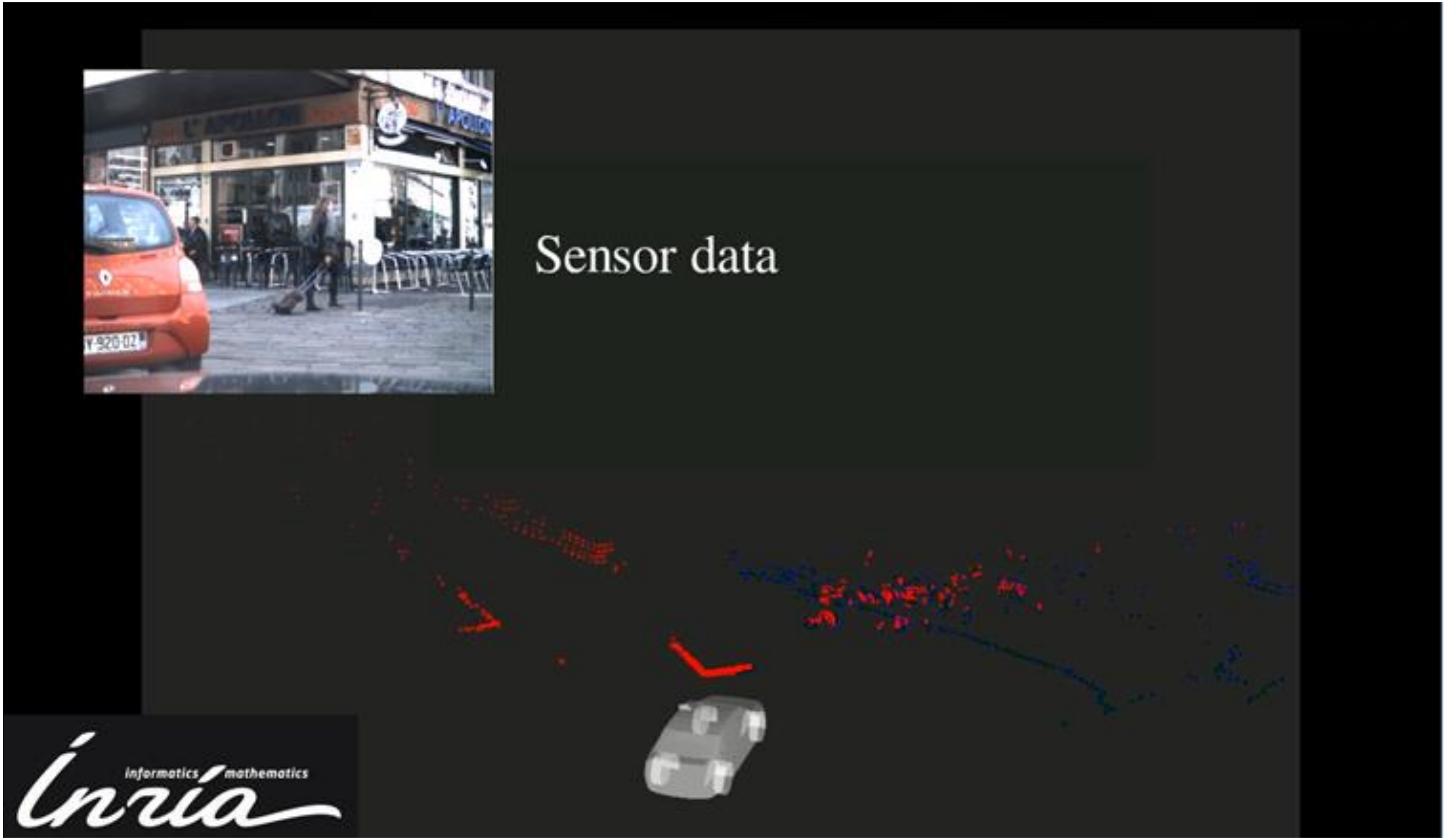
Short-term collision risk – *Experimental results*

⇒ **Detect potential upcoming collisions**
⇒ **Reduce drastically false alarms**



Crash test with no automatic breaking

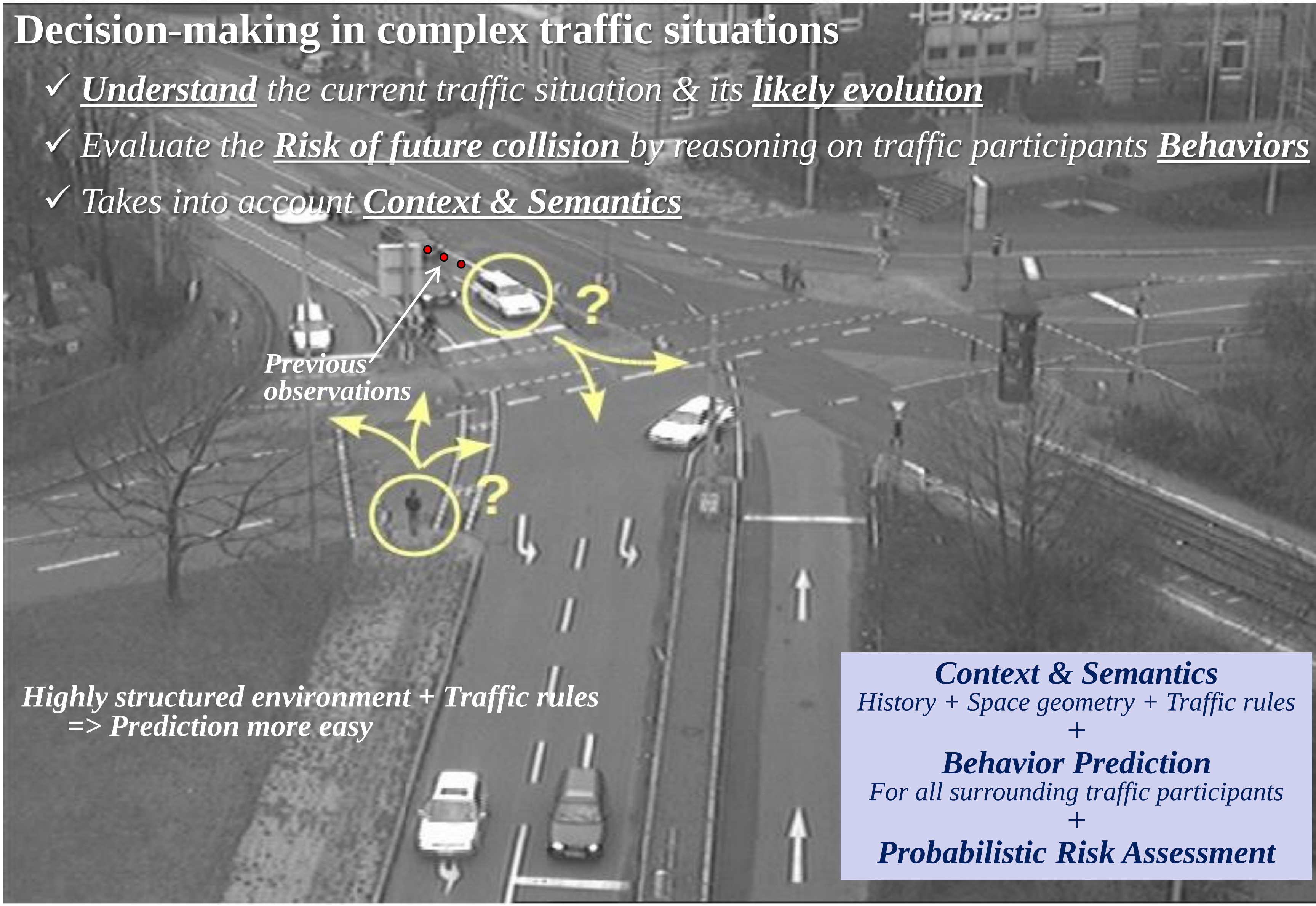
CMCDOT – *Collision Risk Assessment*



Step 2: Generalized Risk Assessment (Object level)

*=> Increasing time horizon & complexity using **context & semantics***

*=> Key concept: **Behaviors Modeling & Prediction***



Decision-making in complex traffic situations

- ✓ ***Understand*** *the current traffic situation & its* ***likely evolution***
- ✓ *Evaluate the* ***Risk of future collision*** *by reasoning on traffic participants* ***Behaviors***
- ✓ *Takes into account* ***Context & Semantics***

Highly structured environment + Traffic rules
=> Prediction more easy

Context & Semantics
History + Space geometry + Traffic rules
+
Behavior Prediction
For all surrounding traffic participants
+
Probabilistic Risk Assessment

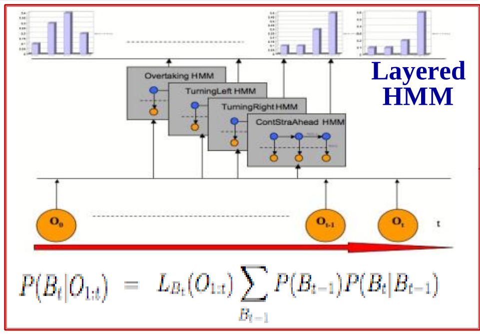
Behavior-based Collision risk *(Object level)*

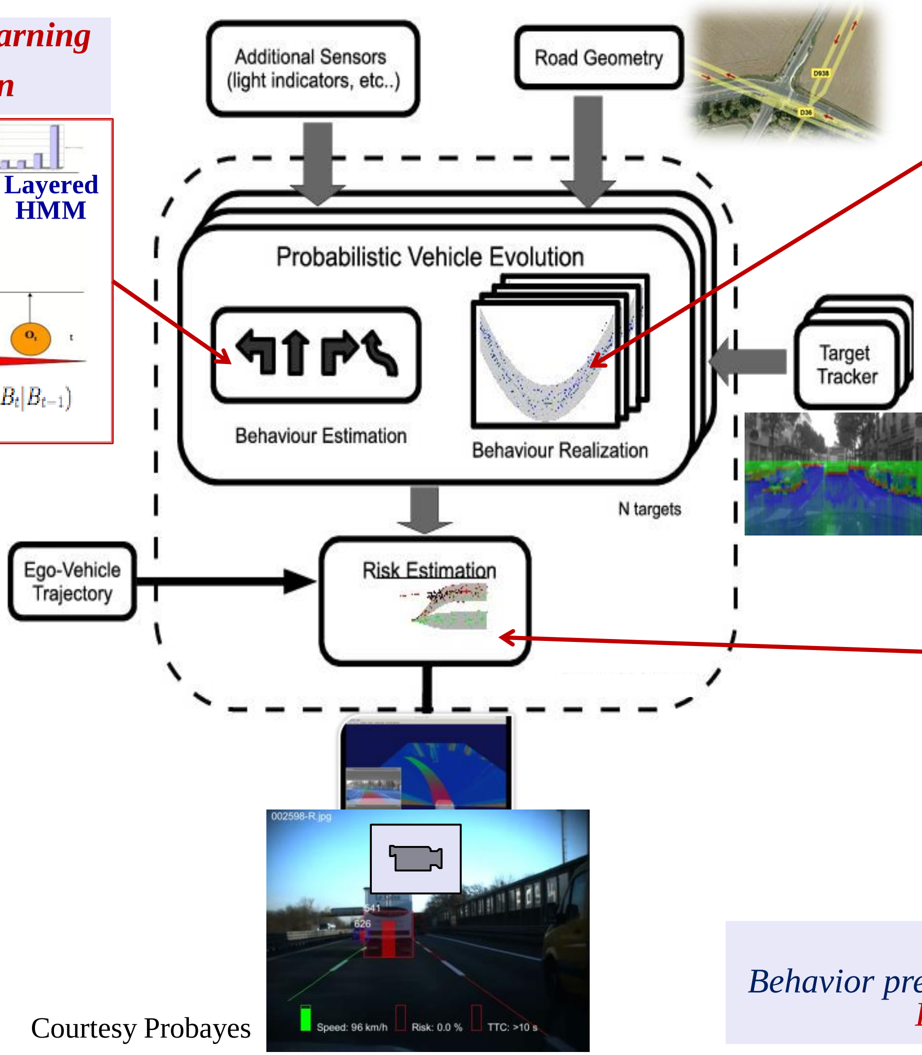
=> Increasing time horizon & complexity + Reasoning on Behaviors

Approach 1: Trajectory prediction & Collision Risk Assessment

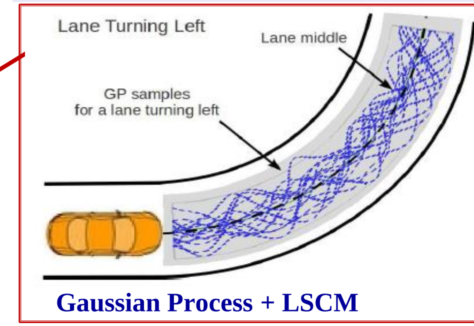
Patent Inria & Toyota & Probayes 2010 + [Tay thesis 09] [Laugier et al 11]

Behavior modeling & learning + Behavior Prediction

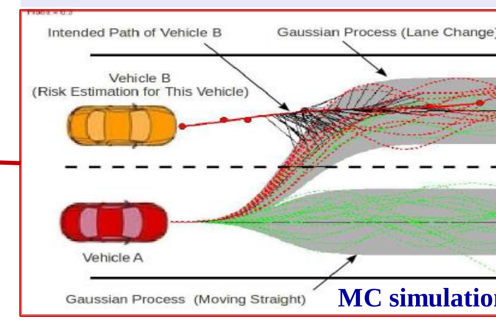




From behaviors to trajectories



Collision risk assessment
(Probabilistic)



Courtesy Probayes

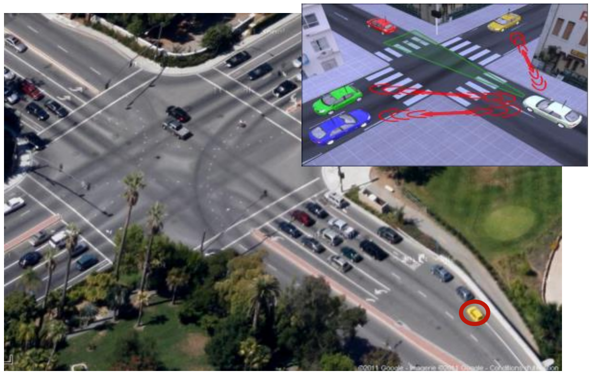
Experimental Results
Behavior prediction & Risk Assessment on highway
Probayes & Inria & Toyota

Behavior-based Collision risk *(Object level)*

=> Increasing time horizon & complexity + Reasoning on Behaviors

Approach 2: Intention & Expectation comparison

*=> Complex scenarios with **Interdependent Behaviors** & **Human Drivers***



[Lefevre thesis 13] [Lefevre & Laugier IV'12, Best student paper]

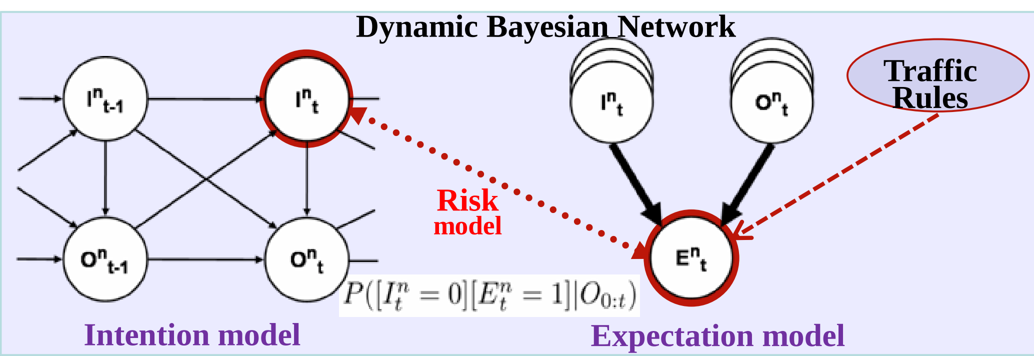
Patent Inria & Renault 2012 *(risk assessment at road intersection)*
Patent Inria & Berkeley 2013 *(postponing decisions for safer results)*

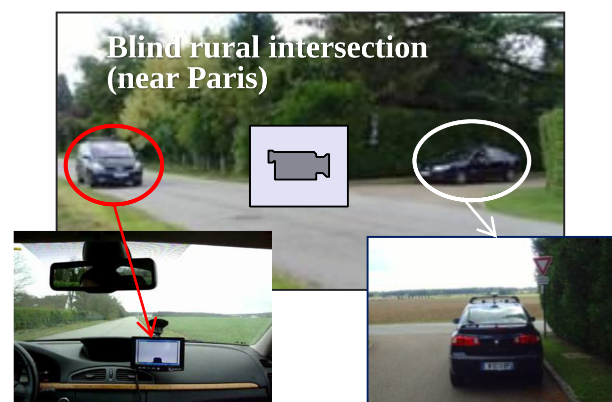
A Human-like reasoning paradigm *=> Detect Drivers Errors & Colliding behaviors*

- ✓ *Estimating **"Drivers Intentions"** from Vehicles States Observations (X Y θ S TS) => Perception or V2V*
- ✓ *Inferring **"Behaviors Expectations"** from Drivers Intentions & Traffic rules*
- ✓ ***Risk** = Comparing Maneuvers **Intention & Expectation***

*=> Taking **traffic context** into account (Topology, Geometry, Priority rules, Vehicles states)*

*=> **Digital map** obtained using "Open Street Map"*



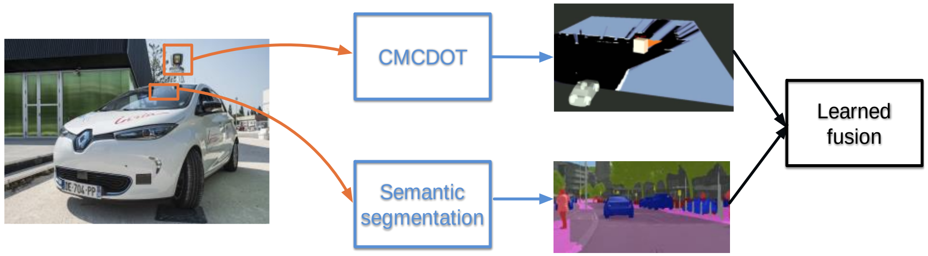


Current & Future work

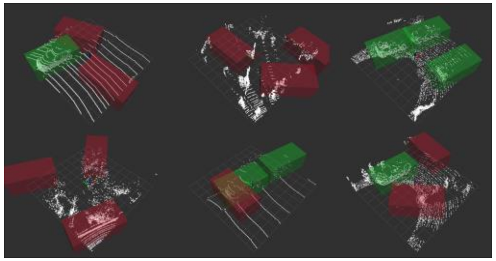
❑ New models/algos for integration in various platforms & dynamics

Integration within the Control Units of EasyMile EZ10 shuttle & Iveco bus & Renault Zoe

❑ News approaches & models for constructing "semantic grids" using CNN

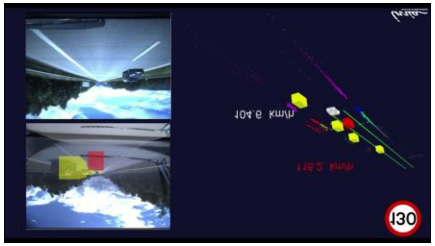


Fusing CMCDOT output (OG) with semantic output from RGB cameras (coop Toyota)
=> Patent application & Publications (IROS 2018 & ICARCV 2018)



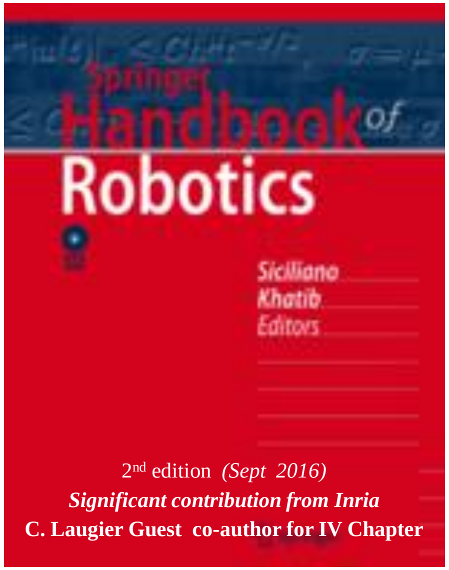
Semantic segmentation in a 3D point cloud
=> Master Thesis

❑ Learning & Predicting driving Behaviors for Autonomous Driving

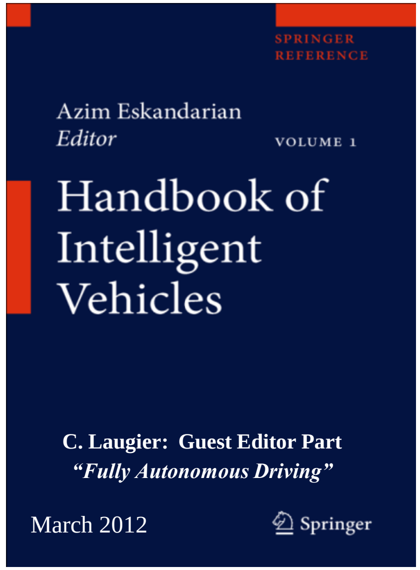


- Driver behavior modeling using Inverse Reinforcement Learning
- Combining Model-based prediction with Dynamic evidence to estimate lane change intentions

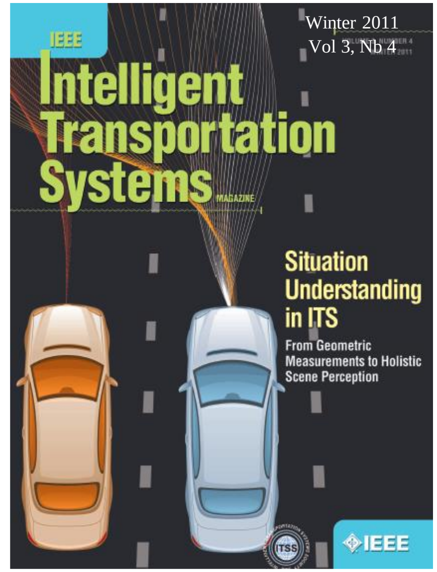
=> 2 Patents application & publications (ITSC 2016, ICRA 2017, ICRA 2018)



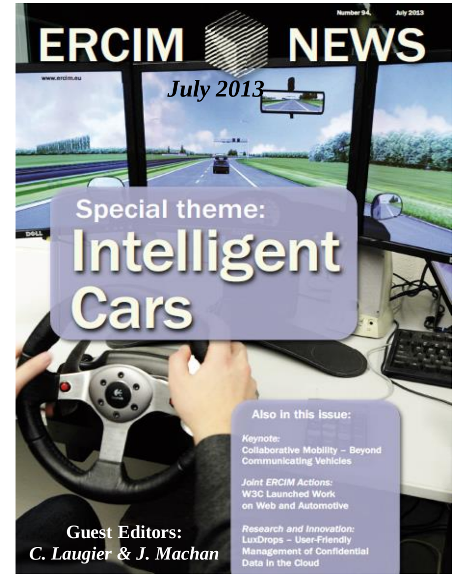
2[nd] edition *(Sept 2016)*
Significant contribution from Inria
C. Laugier Guest co-author for IV Chapter



C. Laugier: Guest Editor Part
"Fully Autonomous Driving"
March 2012

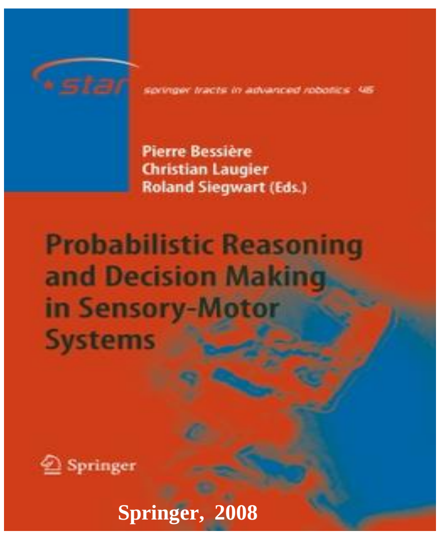


Winter 2011
Vol 3, Nb 4



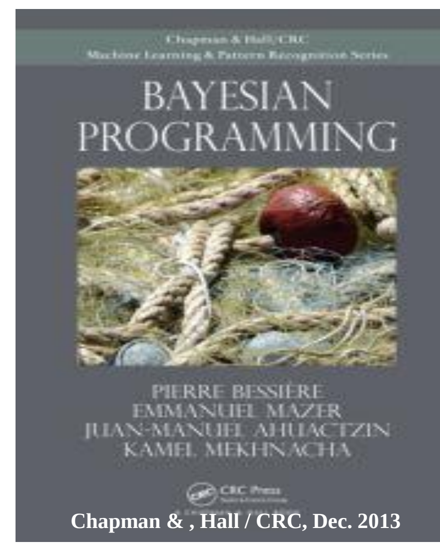
July 2013
Guest Editors:
C. Laugier & J. Machan

Thank You Any questions ?



Springer, 2008

IEEE RAS Technical Committee on "AGV & ITS"
Numerous Workshops & Special issues since 2002
=> Membership open



Chapman & , Hall / CRC, Dec. 2013